Audion Sync Tools — changelog

Русский · User guide

Contents

- 2.5.2 — 28 October 2026
- 2.5.1 — 27 October 2026
- 2.5.0 — 24 October 2026
- 2.4.1 — 23 October 2026
- 2.4.0 — 23 October 2026
- 2.3.1 — 20 October 2026
- 2.2.0 — 18 October 2026
- 2.0.1 — 15 October 2026
- 2.0.0 — 9 October 2026
- 1.19.2 — 26 September 2026
- 1.19.1 — 26 September 2026
- 1.19.0 — 26 September 2026
- 1.18.1 — 24 September 2026
- 1.18.0 — 24 September 2026
- 1.17.0 — 20 September 2026
- 1.16.0 — 20 September 2026
- 1.15.3 — 15 September 2026
- 1.15.2 — 15 September 2026
- 1.15.1 — 15 September 2026
- 1.11.18 — 13 September 2026
- 1.11.17 — 12 September 2026
- 1.11.16 — 11 September 2026
- 1.11.15 — 11 September 2026

- 1.11.14 — 11 September 2026
- 1.11.13 — 11 September 2026
- 1.11.12 — 11 September 2026
- 1.11.11 — 11 September 2026
- 1.11.10 — 11 September 2026
- 1.11.9 — 11 September 2026
- 1.11.8 — 11 September 2026
- 1.11.7 — 11 September 2026
- 1.11.6 — 11 September 2026
- 1.11.5 — 11 September 2026
- 1.11.4 — 11 September 2026
- 1.11.3 — 11 September 2026
- 1.11.3 — 11 September 2026
- 1.11.2 — 11 September 2026
- 1.11.1 — 11 September 2026
- 1.11.0 — 11 September 2026
- 1.10.8 — 11 September 2026
- 1.10.7 — 11 September 2026
- 1.10.6 — 11 September 2026
- 1.10.5 — 11 September 2026
- 1.10.4 — 11 September 2026
- 1.10.3 — 10 September 2026
- 1.10.2 — 10 September 2026
- 1.10.1 — 10 September 2026
- 1.10.0 — 10 September 2026
- 1.9.0 — 10 September 2026
- 1.8.0 — 10 September 2026
- 1.7.0 — 8 September 2026
- 1.6.1 — 7 September 2026
- 1.6.0 — 7 September 2026
- 1.5.1 — 7 September 2026
- 1.5.0 — 7 September 2026

- 1.4.0 — 7 September 2026
- 1.3.0 — 6 September 2026
- 1.2.0 — 5 September 2026
- 1.1.0 — 5 September 2026
- 1.0.0 — 5 September 2026

2.5.2 — 28 October 2026

SSHD and RESTORE check port 22 by connecting. sshd is ready when port 22 of this machine answers with `SSH-`, not when the Windows connection table says so: on Windows 11 26H2 that table returns states at random, and with sshd running the SSHD step and RESTORE MIRROR ended with "port 22 is not listening" (exit=1). Forwards for RDP, the web UI and SOCKS are checked the same way: the program takes a free port for a moment itself and knows a forward is up when it answers.

REVERSE SSH TUN opens no browser. The channel tries a leg through cloudflared only with a Cloudflare Access sign-in; without one it skips that leg and holds on to the others. Sign in with RESTORE MIRROR or `cloudflared access login`.

The Audion logo in ABOUT. ABOUT opens with the Audion logo of Audion Design: light letters in the dark themes, dark ones in the light theme.

2.5.1 — 27 October 2026

A connection without an update of the tools. The probe of the window is tried again while both VPS are silent, even when a machine or a router of this network has already answered: such an answer no longer cancels the retry, and the roads to MAIN VPS and SECOND VPS do not stay "none" until a probe by hand or TOOLS UPDATE. A probe asked while the previous one ran (the network settled after a change, SETUP, an update) goes after it instead of being lost.

SSHD lets in the network of the router at work. `10.12.0.0/24` (WG SECOND VPS ↔ the router at work) and `192.168.15.0/24` (the network behind it) are added to the own networks of the rule for port 22: a machine behind that router is reached through SECOND VPS. SSHD updates the rule.

REVERSE SSH TUN takes the ssh of Windows. When the ssh of Git (`usr\bin`, as in Git Bash) comes first in PATH, the channel passes it by: it cannot read the config of the channel and exited with an error before any connection.

2.5.0 — 24 October 2026

No data of the household in the program. The SYNC ALL plan - MAIN VPS, the R2 bucket, the Forgejo folders on the server, the repositories, the parts of the recovery archive - lives only in `config\mirror.json` of the data folder; `config\mirror.sample.json` of the program is a template with a `<...>` in place of every value. SYNC ALL builds no plan from the program and does not finish an old version 1 plan: without a v2 plan, with a `<...>` left or without `protectedLocal` it stops changing nothing and says what to do.

The names of MAIN VPS in targets.json. The name for the straight road (`host`) and the new name for cloudflared (`cloudflaredHost`) live in the `server` entry; the second is filled in by the field `MAIN VPS: NAME FOR CLOUDFLARED` of SETUP → ADDRESSES. Until the field is there, the probe of the window, the tunnel client and the route check take the names from this machine's `~\.ssh\config`; without a name a route is not written, and the program says what is missing.

SYNC ALL checks what it runs on. Python 3.10+ is looked for as `py -3`, then `python.exe` on PATH (the Microsoft Store stub does not count), and its version is checked; without Git or a fitting Python SYNC ALL says at once what to install. The question before SYNC ALL is "Refresh the library?".

2.4.1 — 23 October 2026

._runtime is disposable in both the program and DATA. SYNC ALL journals and rollback move to DATA/Recovery/sync-all, backup settings to DATA/Recovery, cloud sessions to DATA/config/rclone. Legacy files migrate with hash verification without overwriting conflicting copies. BISYNC requires DATA and never stores state in the public project. Refreshed cloud tokens survive restart; stale sessions cannot undo a newer reconnect. Tests cover deleting both scratch folders, reopening the window, preserving BISYNC and recovering an interrupted SYNC ALL.

2.4.0 — 23 October 2026

SYNC ALL refreshes the complete library, discovers new repositories automatically, and retains every server recovery payload. Checks cover refs/HEAD, Git fsck, SHA-256, the inner manifest, and actual operational file/unit coverage. Validated staging copies are installed transactionally in MIR mode. Server versions replace local edits and extra files. Previous copies exist only for temporary rollback and are removed after success. Keys, connections, recovery snapshots and explicitly protected secrets are not removed. Concurrent local changes stop replacement. CLI and button share one engine; v1 plans migrate safely. Details: SYNC ALL.

2.3.1 — 20 October 2026

The probe tries again by itself. When nobody answers the probe — the program started before the network was up, or a VPN is still connecting — it is tried again in 20 seconds, up to three times in a row. When the network changes (a VPN came up, another Wi-Fi) the window waits for it to settle and probes again. Before, the probe line kept `none` at every route until PORT 22 TCP PING was pressed.

The probe line fits. With SETUP, SETTINGS and ABOUT in the top row it was cut with an ellipsis. The way now follows an arrow, and "(by probe)" is not written — only `chosen` and `by machine` stay in brackets. The whole line is in the tooltip.

Space counts the size of a folder. As in Total Commander: Space on a folder marks it and counts how much it takes — by the disk on this machine, through `rclone size` on a far side. While it counts, the column SIZE shows "…", then the number; it goes into the sum "MARKED" under the panel. "?" — the count had no answer, Space on the folder counts again. Two folders are counted at a time, the rest wait; leaving the folder stops the counts begun. Insert marks without counting; drives in the list of drives are not counted by Space. The exact bytes of a row are still Shift+Space.

The translucency by default is ACRYLIC in the file panels. A fresh copy of the program opens with `ACRYLIC` and "File panels only": the background of the file lists is blurred, the top line, the buttons and the gaps stay solid. A choice already saved in SETTINGS → TRANSLUCENCY stays as it was; with the transparency effects of Windows switched off the window is solid.

2.2.0 — 18 October 2026

SETUP, SETTINGS and ABOUT at the top right. The three buttons stand before the choice of the language and the theme; the row of sections keeps SSH KEYS, TUNNEL, REMOTE, STORAGE and CLOUDS. A tangerine dot lights on `SETTINGS` when a newer version of a tool the program carries is out; the hint of the button names which. `ABOUT` shows the version, the folder of the program and the data folder, and opens the documents, the folder of licenses and the page of releases.

AUTO picks the VPS by its delay. From afar `AUTO` takes the VPS that answers the probe faster: MAIN VPS by its name or SECOND VPS. Less than a tenth apart is no difference, and MAIN VPS comes first then. The route over cloudflared comes after the timed ones. At home — the home network, as before.

The routes per machine — with a kebab in the ROUTE row. The kebab `⋮` at the end of the row opens the routes per machine with the time to the nearest node: `APPLY` — until the program is closed, `MAKE DEFAULT` — for the starts to come as well, `RESET` — `AUTO` for every machine with no defaults. A tangerine dot on the kebab says the routes are not the defaults. A choice made with the

buttons of the row is no longer remembered until the next start: the one remembered before became the default.

The window of the first start chooses the data folder. The folder picked in it with OK was refused with the journal line "an operation is running: the data folder is changed after it", and the program stayed without a data folder: the panels of the machines empty, no indication. Choosing through SETTINGS → DATA FOLDER → CHOOSE worked before as well.

2.0.1 — 15 October 2026

The cryptography packages of PowerShell. Microsoft.PowerShell.SDK 7.6.6 brings System.Security.Cryptography.Pkcs and .Xml 10.0.10, which have known vulnerabilities; the PowerShell of the program (`App\AudionPs.exe`) is now built with version 10.0.13 of both. PowerShell itself stays 7.6.6, and the window does not change.

2.0.0 — 9 October 2026

A new window. The window is rewritten from WPF in PowerShell to C# and Avalonia; the previous one is removed whole. The engine is the same: the scripts of `Engine\Scripts`, `Tunnel\` and the wrappers of `Runs\`, and a button still calls what a console calls. `Start.exe` starts `App\AudionSyncTools.exe`; `--no-admin` skips the question of administrator rights, `-Root "data folder"` names the folder of this run without changing the pointer of the program.

Its own PowerShell 7. `App\AudionPs.exe` is PowerShell 7 inside the program: the scripts of the window, every wrapper and the steps of the builder run under it. Left on the PowerShell of the system are the worker of REVERSE SSH TUN and the relaunch with administrator rights in four installing scripts.

A new mark. The "S" of two opposing arrows is drawn anew, as a vector: sea blue and copper on a transparent ground. Its sources are `Assets\SyncTools.svg` and `Assets\SyncTools.png` (1024); the icon `Assets\SyncTools.ico` is built from the picture by `Engine\Launcher\Build-IconFromImage.ps1`, six sizes from 16 to 256. `Start.exe`, `App\AudionSyncTools.exe` and `App\AudionPs.exe` carry it.

The description of the program is a pair in `Docs`. `Docs\README_RU.md` and `Docs\README_EN.md`, as in the other programs: the Russian text is moved from the root as it was, the English one is written beside it. `README.md` in the root became a pointer to both halves of every pair.

The order of restoring is a pair too. `Docs\RESTORE.md` became `Docs\RESTORE_RU.md`, and `Docs\RESTORE_EN.md` lies beside it. The entries below keep the former name as it was.

The wizards became scripts. What the code of the window used to do is moved into `Engine\Scripts` and works from a console: `Set-Targets.ps1` (ADDRESSES), `New-ClientKey.ps1` (SSH-KEYGEN), `Set-R2Credentials.ps1` (R2 CREDS), `Set-RouterSecret.ps1` (ROUTER PASS), `Build-Transport.ps1` (RCLONE.CONF), `Manage-ClientKeys.ps1` (LIST, SSH-COPY-ID, REVOKE), `Get-Clouds.ps1` (the list of clouds), `Switch-Tunnel.ps1` and `Get-TunnelState.ps1` (the tunnel to the router of BASE). The window asks with a form, a table or a choice; only what holds a conversation of its own opens in a separate console. The hints about SECOND VPS name the `SETUP` section: the `INSTALL` button stands there, not in TUNNEL.

A job ends with its process. A command that started a helper and went kept its job "running" for as long as the helper lived: the window waited for the end of its output with no limit, and `STOP` found nothing to stop. Now the last lines are waited for three seconds after the process has ended, the job gets its verdict, and the queue goes on; the helper is left alone. The builder step `build_app` checks that the new `App\` moved into place and puts the former one back when it did not: a file in use used to leave the program without `App\` under the word "built".

What the previous window did by itself, with no button. A comparison of the new window with 1.19.2 by behaviour found six such places; all are in place. Closing the window during a transfer asks, and on `ABORT` kills the job with its processes and the queue: `rclone` does not outlive the window. The data folder changes in one step: the cloud session of the old folder is saved into it and leaves the window before the folder changes, and the panels open anew; the folder is not changed while an operation runs. The routes of the machines are read from the plain `rclone.conf` also when clouds are attached and the working config is encrypted. COPY, MIRROR and BISYNC between two clouds of one kind run on the side of the cloud — `--server-side-across-configs`. A data folder without addresses gets their frame from the sample of the program at the start, when it is chosen and before the `ADDRESSES` wizard: `Initialize-DataFolder.ps1` does it, and it also brings in the entry of SECOND VPS from `~\.ssh\config` and the working files left in the program folder by versions before 1.15.0.

An action goes with what it began with. While an action counts or asks — COPY counts what will go, F8 waits for its answer, a wizard shows its form — the data folder is not changed: the window says "an operation is running: the data folder is changed after it", as it does during a transfer. A COPY counted in one data folder and confirmed after a change to another would have gone with the paths of the first and the config of the second. An operation goes to the address its question named, even when the panel has gone elsewhere meanwhile — sent by the search window, or opened on WORK after

OPENVPN: F5, F6, F2, F7, F8 and F3 take the place of the panel before the question. The probe of the network, the state and the summary asked of the previous data folder do not land in the new one.

A look with fresh eyes, 8 October. The program was read anew - not by what the window calls `rclone` with, but by what `rclone` then does to the files; what looked doubtful was reproduced with its own build on throwaway folders, on an sftp server of its own and on the machines - BASE and WORK. The number is the same: 2.0.0 has not left the project folder yet. The ten paragraphs below are what was found and fixed; after them - what the audit of these fixes found.

F2 does not rename onto a taken name. `rclone` did it without a word: a file lying under the new name was deleted, folders were merged. The window reads the folder the name goes into first and refuses a taken name; a folder that could not be read refuses it too. Every rename goes with `--ignore-existing`: a file whose name was taken after the question stays in place beside the one that took it, and the job ends with an error and not with "ready". A folder is another matter - see the audit below.

Changing the case of a letter alone no longer costs the file. On a machine with Windows over ssh `rclone` found the old name under the new one, took the source for a needless copy and deleted it: `name.txt` → `NAME.txt` - the file is gone, `dir` → `DIR` - the folder is empty, and the job ended with "ready". On a disk of this machine it would not rename such a folder at all. In both places the case is now changed in two steps by way of a temporary name; in an R2 bucket and in the clouds - in one, as before. `STOP` between the steps does not start the second one, and the journal names what was left undone.

F8 on a disk of this machine goes into the Windows Recycle Bin. A local disk was cleaned the way a server is - for good - while the question spoke of "remote sources". What the bin will not take stays in place, and the job says so. SHIFT+DEL and SHIFT+F8 delete past the bin; the question and its badge say that. A machine over ssh, a server and an R2 bucket have no bin: there everything is as it was, and the question is the old one.

A folder inside a folder is refused before the question. F5, F6, COPY, MIRROR and BISYNC refuse when the target lies inside the source, and a mirror also when the source lies inside the target. `rclone` itself refuses only the plain case. Through a junction, and where one place has two names - a disk of this machine and the same machine over ssh, the way the panels of TUNNEL and REMOTE open on BASE - a mirror deleted the neighbouring folders and the source itself. A copy of a folder into its own subfolder it does not refuse at all and does not stop: eight thousand levels deep in five minutes. F5 and F6 into the very folder they stand in are refused as well.

A pair of BISYNC stays the same pair when the route changes. `rclone` keeps the state of a pair under the names of its two sides, and the name of a machine changed with the route to it. A pair made

at home was a new one from afar: the window asked about it again and went with `--resync` - a change in the target was overwritten, what had been deleted in the source came back. For BISYNC a machine now has one name by any route, and the route itself goes to `rclone` in variables of the process.

VIA MAIN VPS asks and counts. The transfer between clouds started at one press, and it cannot be stopped from the window: the work goes on MAIN VPS by itself. The window now counts how much will go and asks; the question says that `STOP` will not reach it. One transfer runs on MAIN VPS at a time: the log and the file with the process number there are one for all, and a second transfer started before the end of the first wiped both. `-Watch` says "running" only when the recorded number is alive and it is `rclone`.

Esc in a text field stops nothing. Esc in the quick filter or in the box of masks stopped a running transfer together with its queue. In a text field Esc now belongs to the field; from the panels it is `STOP` as before.

A script ends with its own code. Under the window and in a step of RESTORE the code of a script was the code of the last program it had started itself. A script that dealt with a refusal of ssh or robocopy and ended without an `exit` of its own was "code 3" in the window, and RESTORE failed as a whole on such a step. The code is now the one a wrapper gets: the number of its own `exit`, and zero without one. `Update-Tools.ps1` likewise judges the installer of the archiver and the editors by the success of the call, not by a code left by whatever ran before it.

What is downloaded is checked before it is started. The check of a downloaded tool was its start - while the program runs with administrator rights. Before the first start it is now checked whose it is: the signature of Cloudflare for the tunnel client, the signature of Microsoft for Windows Terminal and Microsoft Edit; for `rclone`, micro and PeaZip, which their publishers do not sign, the SHA-256 sum from the file the publisher lays beside the release. A signature or a sum that does not match - not installed, the previous copy stays. A file of sums that could not be fetched, or holds no line for the archive, is said in the journal and the install goes on: the sum guards against an archive damaged on its way. The versions are the latest as before: what SETTINGS shows and what `UPDATE` installs has not changed.

COPY of the journal takes what the panel holds. A second copy of the journal was kept beside the panel and never cut: it grew for as long as the window stood open. The lines of the panel go to the clipboard - the last three thousand; everything since the start of the day is in the file in `logs`.

The audit of these fixes, 8 October. Another chat checked the fixes - by the final code and by trials with the real `rclone`. Eight findings and one observation outside the fixes; the five paragraphs below are what was fixed, under the same number.

The same machine by a far route is the same one place. The window knew its own machine under another name by its address, and on the straight route only. But a button of the row ROUTE sets a far route for the machine the program runs on as well, and a laptop away from home reaches itself through its own REVERSE SSH TUN - and a mirror of a folder into its parent deleted the neighbours again. The machine at the other end is now asked whether it is this one: the window reads its own mark file through it, and only when the paths at both ends match disk for disk. A machine that gives no answer - the operation is not begun.

F2 checks the name once more before the start, and the folder after it. A name free at the press of F2 could be taken while the job waited in the queue behind a long transfer. The folder is read once more right before the start, and the job does not begin onto a taken name. A folder, unlike a file, has no fuse for the moment that is left: `rclone` moves what the folder that took the name did not have into it, leaves what both have, and ends with "ready". After a rename of a folder the window looks whether it stayed where it was, and says so with an error. F2 waits for the folder as long as the panel does - a minute, not half of it. When the second step of a change of case fails or is stopped, the journal names the temporary name the object is left under.

A panel does not keep the rows of the folder it left. A disk of this machine showed the rows of the previous folder until the list of the new one came, and F2, F8 and F5 took a name from the old list and the path of the new folder in that moment. The rows are cleared at once, as for a machine over ssh. The window asks the disk about nested folders off its own thread: a mapped drive that stopped answering no longer holds it.

A warning of rclone is hidden in one job, not in the whole journal. The line about ssh over a host is expected only in BISYNC from afar, where the window does that itself; in any other job it tells of a mistake in the working `rclone.conf` and is seen again.

A script looks at the refusal of a program its report depends on. The code of a script being its own now, a refusal of `icacls` or `git` no longer surfaces in the window by itself. A script checks it where it would otherwise report success: the permissions of a key in EXPORT KEYS (a refusal is an error of the script), in SSH-KEYGEN and in the install of OpenSSH, the fix of `core.sshCommand` in IMPORT KEYS, the removal of a cloud. The sum of PeaZip is looked up by the name its publisher gave the file when the archive came by the fallback road as well.

The first start. A release comes without the tools. The window of the first start asks for the data folder and downloads what is missing, about 100 MB: `rclone`, `cloudflared`, Windows Terminal, PeaZip, the editors Microsoft Edit and micro. It comes back at every start while a tool is missing.

Masks. The asterisk button between the panels marks rows of the active panel: masks of your own separated by commas, saved sets, the time of change, and groups of extensions to include or exclude.

The search. Ctrl+F, Alt+F7 or the magnifier between the panels: from the folder of the active panel downwards, on a disk, on a machine over ssh, in an R2 bucket and in a cloud. It searches by a part of a name, an extension, a mask, a regular expression, groups and time. What is found opens in the panel as a flat list, or by going to the file.

Keys. Added: Tab, Alt+← and Alt+→ (the history of the panel), Alt+F1 and Alt+F2 (the list of sources), F (the quick filter), Ctrl+A and `*` (marks), Shift+Space (the exact size), Ctrl+R, Ctrl+U, Ctrl+Enter (a terminal where the panel stands), Ctrl+Shift+N and Ctrl+Shift+P (names and addresses into the clipboard), SHIFT+DEL and SHIFT+F8 (delete past the Recycle Bin). Esc stops the running operation — from the panels; in a text field it belongs to the field.

F3, F4 and archives. F3 puts into the clipboard the public links to what is marked in an R2 bucket that has a site of its own. F4 opens a file in the editor, Shift+F4 creates a new one; the editor is chosen in SETTINGS. Alt+F5 and Alt+F9 pack and unpack archives through PeaZip: ZIP, 7Z, SFX, TAR, TAR.GZ, TAR.ZSTD, TAR.LZ4, with a password and a check of the archive. The editor and the archives work on the disks of this machine.

Indicators without red. The setting-up buttons have three meanings: grey — it is not there, amber — it is there but not proven in practice, green — proven by a panel that was read. The state comes from `Get-SetupState.ps1` and `Get-BeaconState.ps1` as one line of JSON; it is read at the start, after the queue of jobs, after a wizard and by the `STATE` button. In the log red is left only for what broke. Which key is this machine's the indicators find out by the data folder and the profile, not by a name the window remembers: a fresh copy of the program shows the key in place as soon as the data folder is named.

An error in a handler does not close the window. It goes whole into `logs\gui-crash-<time>.log`, one line with the name of the file stays in the log, and the window works on. Twenty errors within a minute — the program closes, and the last file tells why. Errors of background tasks nobody waited for are written to `logs\gui-background-<date>.log`.

The summary of the state — by a script of its own. `STATE` and the start of the window write the summary and the report on the tunnels into the log, as before; the facts now come from `Get-StateSummary.ps1`. A tunnel client that is gone shows for a tunnel that is down too: the check used to run only for one that was up.

The sets of masks and the pairs of BISYNC — in the data folder. The sets lie in `config\mask-sets.json`, the pairs in `config\bisync\pairs.json`, and what `rclone` keeps for each pair beside them, in `config\bisync\state`. The list of the pairs used to live in the settings of the window and the state in `._runtime` of the program: a cleanup took the state and left the list, `rclone` refused with "must run --resync", and the window could not ask again. Now a pair is known while its state is

there. The pairs of earlier settings are not carried over: the first run of each asks again, and the log says so.

SETTINGS. A new section in place of the panels: the data folder, the tools and their versions (`CHECK VERSIONS` only asks, `UPDATE` handles all six), the editor of F4, the size and the choice of fonts, the translucency of Windows 11, sizes in bytes, the bottom row without captions. There are four themes: Audion Vault Dark, Audion Dark, Audion Code Dark, Audion Light.

The strip between the panels. The power button of the tunnel to the router of BASE, the routes by machine and the pin of a section moved there; next to them stand masks, the search, the swap of the panels, and the names and full addresses of what is marked into the clipboard.

Row colours. They are read from `config\colors\themes.json` and `config\mask-groups.json`: the groups of the masks, of the search and of the colours are one list. The Total Commander schemes `config\colors\*.ini` and the `Update-AudionColors.ps1` script are removed from the program.

The golden copy of the clouds is taken from the data folder the window works with. After a cloud was edited, `Manage-Clouds.ps1` uploaded to MAIN VPS the config of the folder the pointer names, even when it was told another one: `-Root` did not reach the upload.

Checks. There are 29, all running under the built-in PowerShell through `Tests\Run-Tests.ps1`; a name that fits no check is an error now, and the last line says how many passed. New: `Test-NewWindow.ps1`, `Test-ClientKeys.ps1`, `Test-StateSummary.ps1`, `Test-DataFolder.ps1`, `Test-ScriptExit.ps1`, `Test-ToolTrust.ps1`. `Test-Cli.ps1` and `Test-Prompts.ps1` read the new window. `Test-Window.ps1`, `Test-Handlers.ps1` and `Test-TransportRoutes.ps1` left with the previous window: the logic they checked is C# now. Its behaviour is tried by the stand, `Tests\Stand\Run-Stand.ps1`: the built window with no screen and no mouse, stubs in place of PowerShell and `rclone`, eighteen probes — a question is left open, the data folder is changed or a panel is sent elsewhere, and only then the question is answered. The stand needs the dotnet SDK, takes about two minutes and is not a part of the common run.

The engine. The functions the scripts stand on live in `Engine\BackupEngine.psm1` — with no window in it. Five functions nobody called are removed. The window has one version number, `config\version.json`: the build project reads it from there and has none of its own.

Documents. `README.md` and both guides are rewritten for the new window.

Not tried live: the talk with a real router, the installer of WireGuard, raising and switching a tunnel of our own, the sign-in to a cloud through a browser, SSH-COPY-ID and REVOKE on real machines.

The rims of the controls are quieter, the coloured buttons of the bottom row have no fill, 10 October. The rims of buttons, switches, check boxes and fields - at rest, under the pointer and of the ones that are on - keep 40% of their strength: many rims side by side flickered. The rims of panels, cards and plates and the separators are as they were. F5, F8, F11 and F12 of the bottom row lost their coloured fill: the button stands on the ground of a plain one, and the colour of its rim speaks for it. The number is the same.

1.19.2 — 26 September 2026

The red SFTP column of RESTORE MIRROR is solved. The first MIRROR on BASE with 1.19.1 named the cause: `Invalid value when setting --s3-no-check-bucket from environment variable RCLONE_S3_NO_CHECK_BUCKET=""`. After every rclone call the program put the environment back as it was, and "put back" a variable that had been absent by handing `$null` to `SetEnvironmentVariable`. PowerShell turns that `$null` into an empty string, and PowerShell 7, where the window lives, keeps an empty string as an EMPTY variable instead of removing it. So after the first rclone in the window every process started from it - RESTORE, the external windows - got `RCLONE_S3_NO_CHECK_BUCKET=""` and `AWS_SHARED_CREDENTIALS_FILE=""`, and its rclone died at start. A separate run of the same checks did not get these variables and passed. Removal now goes through `[NullString]::Value` in all 17 places: the restore after rclone, the clouds password `RCLONE_CONFIG_PASS` in the window and in `Manage-Clouds.ps1`, the cloud remote variables. The same explains the old case of a panel failing on "an empty type" of a cloud. The new test `Tests\Test-EnvRestore.ps1` checks both the code and a live call under PowerShell 7.

1.19.1 — 26 September 2026

The route check names the reason. The SSH/SFTP table at the end of RESTORE no longer prints a bare "FAILED / TIMEOUT": lines under "Why:" give each failure its reason - the last error line of ssh or rclone, the exit code, or "no answer in N s". A probe that failed fast is asked again alone before the route is called dead; if it passes the second time the route is OK and the first reason stays in the line. The cause was the SFTP column of MIRROR on LAPTOP on 26.09.2026: every probe failed at once while the routes were alive, and the table threw the reason away. `Invoke-AbkHostScanBatch` gained `-Detailed` (output, errors, exit code, timeout); SSH-KEYSCAN calls it as before.

The window no longer hangs on cloudflared. The network probe counts the route through cloudflared as alive only when port 443 of `ssh.audion.dev` is open and there is a Cloudflare Access sign-in (`cloudflared access token`, in the probe's background thread, 5 s at most). Without the sign-in the route is silent, the window takes the next one, and the log gets one line on how to sign in.

Before, cloudflared opened a browser and waited for the sign-in with no time limit, ssh waited for it, and the window stood still.

The build no longer opens a terminal. Installing Windows Terminal checked itself by running `wt.exe` with no arguments, which is simply "open a window": every build and every tools update left an empty PowerShell on the screen. Now the terminal is checked by its file and version, without being started.

1.19.0 — 26 September 2026

RESTORE → "Clean from DATA (MIRROR)". The previous RESTORE only filled gaps: identical keys stayed, an existing `~/.ssh/config` was kept, foreign lines in `administrators_authorized_keys` survived. On 25.09.2026 BASE kept its old 8-host profile instead of 20 hosts after such a RESTORE, and half of the rclone routes did not open. MIRROR makes C: exactly what DATA describes: `~/.ssh` is rebuilt as a whole, the administrators key file holds exactly CLIENTS, SSHD host keys come from `Tunnel\SSHD` even over different live ones, and an old `rclone.conf` outside DATA, REVERSE SSH TUN shortcuts and the `Audion Backup Sync` task are removed. Nothing is deleted silently: everything removed goes to `DATA\._runtime\mirror-<time>` with a closed ACL, together with the `restore.log` transcript. A failed import puts the previous `~/.ssh` back. Human steps wait without timeouts: a declined UAC prompt is asked again, the Cloudflare Access sign-in is verified and retried, and a router password is saved only after a real login. Whether all routers share one password is the owner's choice: asked once and kept in DATA (`targets.json`, `_routers.sharedPassword`). Without it one router's password is never tried on another; ROUTER PASS in the window follows the same choice.

The RESTORE SFTP check no longer rewrites `rclone.conf`. An on-the-fly `:sftp:` remote detected the remote shell and saved the config; seven parallel checks on Windows got in each other's way, and on 26.09.2026 every SFTP row failed while the routes worked. The shell type is now given explicitly.

SSH-KEYSCAN collects what a clean profile needs. MAIN VPS is also written under `audion-vps`, the `HostKeyAlias` both of its routes check. The BASE router is scanned: FILES ROUTER never opened without its key. The ssh fallback checks the key under the target's own name, since Windows `ssh-keyscan` 9.5 cannot finish `sntrup761` KEX with MAIN VPS. The scanner's `-T` is shorter than the batch wait: the Windows scanner held a connection until `-T`, and LAPTOP and the router were killed together with their keys. DSA is no longer accepted; it made direct scans report a false "HOST KEY CHANGED".

The router address comes from DATA. `Check-Router`, `New-RouterPeer` and ROUTER WEB used a hard-coded `192.168.1.1`, while home has long been `192.168.10.x`. The address now comes from the router entry in `targets.json`.

ROUTER PASS with two routers. The router list reached the chooser as a nested array: invisible with one router, with two the choice was skipped and saving failed on `-Router`. The window still said "the password is stored" although no file was written.

1.18.1 — 24 September 2026

A build through the orchestrator finds PowerShell. The orchestrator cuts a step out of `builder_main.cmd` and runs it without the file's header, and only the header looked for PowerShell. On 24.09.2026 all three install steps — `install_rclone`, `install_terminal`, `install_cloudflared` — printed `No PowerShell found`, and the program was left without rclone, the terminal and the tunnel client. These steps and `tests` now look for PowerShell themselves, in the same order as the header. From the menu the builder worked before as well.

1.18.0 — 24 September 2026

F11 COPY now asks first. There was no confirmation here: COPY deletes nothing, so the question looked redundant. But F11 sits next to F12 and gets hit by mistake — on 24.09.2026 that sent 16,754 files, 4.6 GB, in 25 seconds, silently. The dialog names both sides in words, the scope (the whole folder or the ticked items only) and, on a badge, how much would go: files and size. The number comes from an rclone dry run with the same keys and filters as the transfer itself, so it answers the question asked rather than a similar one. When the target is already current, the badge says so: nothing to copy. If the count fails, the dialog still asks, only without the number.

1.17.0 — 20 September 2026

Reinstalling Windows without a round of the machines. SETUP → SSHD saves the service's host keys into `Tunnel\SSHD` of the selected DATA together with a manifest, and SETUP → RESTORE puts them back before sshd is installed and first started. The machine keeps its identity, and its neighbours do not have to trust it again. RESTORE also imports the keys, repairs paths, brings back the Startup shortcut of an enabled REVERSE SSH TUN and prints an SSH/SFTP table. A live key that differs from the saved one is not overwritten; a damaged copy is not applied. Console `ssh` and ProxyJump get a `UserKnownHostsFile` with the file from DATA. The order is in `Docs\RESTORE.md`, the console door is `Tunnel\RemoteDesktop\restore-machine.cmd`.

PRIVATE BACKUP no longer displaces a fuller snapshot. Before an earlier snapshot is removed its contents are compared: a snapshot holding a project the new one lacks is kept, and the program names the project that keeps it. Removal goes object by object, without probing bucket versioning. `-Keep` accepts values from 1, `-ProjectName` takes a comma-separated list. The MAIN VPS mirror removes only its own earlier snapshots from `Snapshots`, by name, instead of everything that lacks today's date.

IMPORT KEYS. A file identical to the golden copy is skipped; a refusal on one key is reported as `failed:` and does not stop the rest.

`-NoPause` in the wrappers. Wrappers in `Runs\` and `Tunnel\` skip their own pause when the switch is given: a finished background run no longer waits for a key. Without it the pause stays, so a window opened by double click does not vanish with the error.

The build pair. `cleanup_project.cmd` takes the program folder back to its sources: it removes the downloaded `Tools\rclone`, `Tools\terminal`, `Tunnel\Bin`, logs, `._runtime` and this machine's state - `config\profile.json` and `config\gui-settings.json`. `Start.exe`, the rest of `config` and the data folder beside the program are left alone; after a cleanup the window asks for DATA FOLDER again. `builder_main.cmd` brings the tools back with `install_rclone`, `install_terminal` and `install_cloudflared` - the same script the UPDATE buttons run - and carries `tests`, `verify` and `clean_cache`. The tunnel client is fetched without `Install-TunnelClient.ps1`: a build only downloads and never edits the ssh profile.

Terminal settings from `config\terminal`. After the terminal is installed or updated, the missing files from `config\terminal` - the `.portable` marker and `settings\settings.json` - are placed into `Tools\terminal`. Settings already in use are not overwritten, and the step runs even when the terminal version is already current.

1.16.0 — 20 September 2026

BASE networks after the move to `192.168.10.0/24`. The former `192.168.1.0/24` is gone from the trusted networks of the `Audion Remote Files (SSH 22)` rule - in SSHD and in the fallback OpenSSH install alike: both scripts create the same rule and now share one list. The default network of the `WG BASE-ROUTER` wizard with an empty `routes.txt` is `192.168.10.0/24`, in the window and in `setup-base-vpn.cmd`. Machines pick the new rule up on the next `setup-ssh-server.cmd` run.

PRIVATE BACKUP stores the snapshot under `audion-backup/Projects`. The default R2 path changed from `builders` to `Projects`; `-Remote` still overrides it.

1.15.3 — 15 September 2026

REVERSE SSH TUN after the setup console closes. The GUI restores the configured worker directly and retries unexpected worker exits after 30 seconds without registering the key again. STOP and removal are respected; old LAPTOP tunnels are not restored. Closing the GUI terminates its worker and SSH children without orphans. Startup errors and OpenSSH progress are logged; the green indicator requires the server to acknowledge the forward. SSH connection establishment is limited to 15 seconds per attempt.

1.15.2 — 15 September 2026

SSH-KEYSCAN keeps the window responsive. GUI and `Runs\ssh-keyscan.cmd` use the same separate worker. Hosts are checked in parallel: up to five seconds for direct scans and another five for configured SSH routes. STOP terminates the worker and its descendants. Unreachable hosts keep their previous keys; changed keys require explicit acceptance with Ctrl or `-Force`.

Private SSH key permissions after migration. IMPORT KEYS repairs inherited and explicit foreign permissions, including files already in place. An incomplete Export Keys is supplemented with server pairs from `Tunnel\SSH\servers`, without automatically importing neighbouring machines' private CLIENTS keys. During REVERSE SSH TUN setup, `bad permissions` triggers ACL repair of the selected IdentityFile within machine data or the profile, followed by another login attempt. Key bytes are preserved; inspection and removal do not change ACLs. The client installer receives the GUI's selected DATA folder explicitly and checks the permissions of the server keys it uses.

1.15.1 — 15 September 2026

First VPS connection during REVERSE SSH TUN setup. An empty `known_hosts` now leads to OpenSSH's fingerprint confirmation in the separate window, followed by a retry in strict batch mode. Changed fingerprints are refused. Inspection, removal and redirected input never request confirmation. Errors from every route are retained in a setup log, so a MAIN VPS error no longer hides why SECOND VPS failed. `SSH-KEYSCAN` includes SECOND VPS and its SSH port. Synthetic first-run and failure checks cover PowerShell 5.1 and 7.

1.11.18 — 13 September 2026

The router password is stored portably, by router name. It used to be sealed with DPAPI - bound to the account and the machine, which broke on every Windows reinstall and did not travel with the data

set: of the whole set only the router password failed to survive a restore, and on the portable drive it was dead weight. It is now written base64 to a file named by the router (`router-base.secret`, `router-home.secret`) - it survives a reinstall and travels, like the keys and the cloud password. By router name, not by address: the home and village routers share the address `192.168.1.1`, and per-address files clobbered each other; now different routers get different files, and their passwords may differ. The file is useless without access to the router - the lock is the folder permissions and the medium encryption. The router scripts (`new-router-peer`, `check-router`, `check-routes`) gained a `-Router` switch. The old DPAPI `router_<address>.secret` is orphaned - the password is entered once with ROUTER PASS.

1.11.17 — 12 September 2026

The log stops shouting about the expected. An unreachable machine (a panel reaching BASE from another network) no longer drops an `[failure] ... ErrorActionPreference Stop` crash into the log: `Invoke-AbkRcloneCommand` now swallows the throw and returns the exit code as its contract promises — the window runs under `Stop`, and in PowerShell 7.4+ a non-zero rclone exit otherwise bubbled to the global handler. Unreachability is now one plain line ("BASE: not answering"), at most once every 15 minutes, instead of a wall of rclone output. The repeating "taking longer than five seconds, waiting" note is gone. For a stale transport (a foreign drive letter in `rclone.conf`, `known_hosts` not found) the message is a clear "transport not built, press RCLONE.CONF".

1.11.16 — 11 September 2026

The dialog-string check catches untranslated text again. In the previous version the check itself hit the classic PowerShell trap: `$matches[1] -match` overwrote the automatic `$matches`, and the remainder of a "label: value" line was lost - the test stopped noticing an untranslated half and silently passed it. The groups are now saved into variables before the next match. A self-check was added: on a synthetic "label: rest" with two deliberately untranslated halves the check must return both. Found by the 1.11.15 recheck (a P3 in the test; the application is unaffected).

1.11.15 — 11 September 2026

Dialogs assembled before translation now read through in English. The "Delete ...?" heading substituted the name into the template before translation, so the finished string without `{0}` matched no pair; the body of the transfer-abort warning was built by interpolation and was not translated at all. Both are now translated as a template with the value inserted afterwards, not before. A pair was added for the CONTINUE button, which also stayed Russian. Test-Window now checks `-`

`CancelText`, catches interpolation with Cyrillic and a Russian `-f` template not wrapped in translation. Found by the review recheck (the remaining P3). A duplicated 1.11.13 heading in the log was removed.

1.11.14 — 11 September 2026

Removing the tunnel key can no longer empty the file and report success. The removal script ran without `set -e`, so after any file operation failed - the filter, the replace, the permissions - it still printed `removed`: an empty `authorized_keys` holding other machines' keys looked like a clean removal, and the holder above cleared the state. Removal now works like registration: a copy before the edit, a line-count check before the replace, and any failure aborts the script before it answers - the file is intact and the state is kept for a retry. A read failure and "no key" are now different answers. Found by review 17967bb (P1); the scenario was added to Test-Beacon.

The window dialogs now read through in English. Headings, buttons and body lines of `Show-AbkDialog` are translated inside the dialog, and a couple of dozen pieces (the labels NOW, FILE, TARGET, ALIGN and others, buttons, headings) had no pair - the English window showed them in Russian. The rest of a "label: value" line is now translated too, and the missing pairs were added. Test-Window now checks the strings of every dialog, not only `Get-AbkText` calls.

1.11.13 — 11 September 2026

The guides now explain the tunnel account. Only SKILL.md and the changelog knew about `tunnel`, while since 1.11.12 the button can answer that the server has no such account. Both guides now say what the account is, why the tunnel key lives there and where its server-side limits come from.

1.11.12 — 11 September 2026

The REVERSE SSH TUN key lives in an account of its own. It used to be written for the working user on MAIN VPS and SECOND VPS. The tunnel key sits on a machine in someone else's network, and a stolen one reached everything that user can reach: files, groups, service sockets. The tunnel now has its own `tunnel` account - no shell, a locked password, nothing in its home but the key file. What the server allows it ships as `Tunnel\RemoteDesktop\config\sshd-tunnel.sample.conf`: a reverse forward on the loopback and nothing else.

The account is created by hand once per server. Until it exists, raising a tunnel breaks nothing and says what is missing.

1.11.11 — 11 September 2026

The English window: code strings. 24 strings (25 calls) that the window translates with `Get-AbkText` right in the code had no pair and showed in Russian in EN: the cloud sign-in explanations, the SSH-KEYSCAN host key change warnings, the lines about the BASE router tunnel, the stream count. They have pairs now, and `Tests\Test-Window.ps1` checks such literals too. Finding N6 of review 2035941.

Setting-up indicators in the guides. The tables said "green when the file is in place", while the window paints amber until the thing is confirmed: R2 CREDS until an R2 bucket is read in a panel, SSH-KEYGEN and IMPORT KEYS until a machine or VPS panel is read, SSH-KEYSCAN and RCLONE.CONF until a remote panel is read, ADDRESSES until a machine answers the probe. The tables now have two columns: when the indicator is lit and what turns it green. Finding N7.

Test-Prompts on a clean copy. The prompts name `Start.exe`, which is built from sources and not kept in git, so the test failed on a clean copy. For `Start.exe` its builder `Engine\Launcher\Build-StartExe.ps1` is now enough. Finding N8.

1.11.10 — 11 September 2026

The router password lives in the data folder the window and the wrapper work with. `Check-Router.ps1` and `New-RouterPeer.ps1` had accepted `-Root` since 1.11.4, but the `RouterNdm.psm1` functions never got it: the password was stored and read in the folder from the `config\profile.json` pointer or the neighbouring one, even when the window had chosen another. `-Root` now passes through `Save-RouterSecret`, `Get-RouterSecret` and the move of the old file - from those scripts, from `Check-Routes.ps1` and from the window (ROUTER PASS, reading the router peers). Finding N3 of review 2035941.

The SETUP → TOOLS → ROUTER PASS indicator sees a stored password. The window called an internal module function that the module does not export: the call failed, and an empty `catch` left the indicator grey. The module now exports `Test-RouterSecretStored`. Finding N4.

Tests. `Tests\Test-Cli.ps1`: every call of a function that looks for the data folder passes `-Root` - in scripts that declare the parameter, and in the window for the router password functions; a synthetic password lands in the `-Root` folder and the indicator sees it.

1.11.9 — 11 September 2026

REVERSE SSH TUN no longer removes a neighbour's key. Registering and removing the tunnel key on MAIN VPS and SECOND VPS looked for the `audion-beacon-<name>` mark as a substring: machine `lab` took the key of `lab2` along with its own, and the line count check did not notice. The line is now recognised by the whole key comment. Finding N1 of review 2035941 (P1, older than this review).

The tunnel key is limited to its own port. On the VPS the key was written as `restrict,port-forwarding,command="/bin/false"`, and `port-forwarding` allowed `-L` to any address the VPS can reach and `-R` on any port. The line now carries `permitlisten` and `permitopen` for this tunnel's port. Keys registered earlier get the limits on the next raise: TUNNEL → LOCAL → REVERSE SSH TUN → ПОДНЯТЬ. Unix-socket forwarding cannot be closed by key options - that is `AllowStreamLocalForwarding` in the VPS's own sshd. Finding N5 (P2).

The key passphrase check no longer hangs. `Test-AbkKeyOpens` called `ssh-keygen -y` with closed input, but ssh-keygen asks the console for the passphrase: on a locked key the check waited until the process was killed - this is how Test-Beacon hung. The empty passphrase is now passed explicitly (`-P ""`), the exit code counts, and the check gets 15 seconds. Finding N2 (P2).

Tests. `Tests\Test-Beacon.ps1`: a locked pair is recognised in under 10 seconds; the real registration and removal scripts run in Git Bash on a synthetic file with `lab` and `lab2` keys; the key line is limited to its port.

1.11.8 — 11 September 2026

The last old names. `Setup-BaseVpn.ps1` called the BASE router tunnel wizard "ROUTER-BASE WG" in its help, its header and the "the tunnel is raised with the button..." hint - now `WG BASE-ROUTER` and the path TUNNEL → LOCAL → WG BASE-ROUTER. The R2 CREDS log header reads "R2 CREDS" instead of "КЛЮЧИ R2". The `F12 SYNC →` confirmation and its run title used the old "ЗЕРКАЛО" - now "SYNC".

1.11.7 — 11 September 2026

Second wording pass: the documents. README, SKILL, USER_GUIDE RU and EN, PROMPT RU and EN, CLAUDE and AGENTS and the machine access README use the same terms as the window and the scripts: golden copy, Export Keys, CLIENTS, public and private key, R2 bucket, jump host and ProxyJump, indicator, rclone.conf, REVERSE SSH TUN.

The documents caught up with the window. The setting-up table describes the current buttons of the SETUP and SSH KEYS modes and when each indicator turns green; the `КЛЮЧИ R2` and `СКЛАД` buttons, the `ИНСТРУМЕНТЫ` group and the "SETUP → TASKS: the mirror then updates itself every night" step have long been gone from the window. The prompts and SKILL name the mode of every step. The machines are described in the REMOTE mode, the probe button is `PORT 22 TCP PING`, the tunnel wizard is `WG BASE-ROUTER`.

Inaccuracies fixed. The English guide claimed the window does not ask for administrator rights - it asks once. SKILL promised that the cloud password arrives with import-keys.cmd, and the guides that `rclone.conf` travels with the key set. SKILL listed `500/udp` for beacons on SECOND VPS that were removed on 09.09.2026. In PROMPT_EN step 5 repeated itself as a torn fragment, and the beacon paragraph described the old WG tunnel. The guides said `clients` holds public keys only - it holds pairs.

The SETUP → TOOLS → ОБНОВИТЬ tooltip promised a green indicator "when all three are in place", while the window checks the tunnel client and `rclone`; it now says so.

1.11.6 — 11 September 2026

Second wording pass: scripts, their help, wrappers and tests. Script messages, `Get-Help` text, `rem` lines in the wrappers and test titles use the same terms as the window: Export Keys, golden copy, CLIENTS, public and private key, R2 bucket, host, Endpoint, SSHD. Internal code comments were not touched in this pass.

The cloud password. Sync-CloudKeys and `clouds-pull.cmd` advised running "import-keys.cmd first" on a new machine, but EXPORT KEYS and IMPORT KEYS do not carry `Tunnel\clouds.key`. They now say what is true: the file is put into the data folder by hand. The Manage-Clouds help and the `Clouds.ps1` header no longer claim that `rclone.conf` travels with the key set.

Manage-Clouds without rclone pointed to the УСТАНОВИТЬ button, which installs only cloudflared; it now points to SETUP → TOOLS → ОБНОВИТЬ.

The old program name. Export-AudionKeys, Import-AudionKeys, Sync-AudionBackup and Backup-AudionArchives printed "AUDION BUILD BACKUP" in their headers; now it is "AUDION SYNC TOOLS" and the button name.

The window log colours EXPORT KEYS and IMPORT KEYS output by the new wording: "not in Export Keys" and "public key only" as warnings, the `export :` label as muted.

1.11.5 — 11 September 2026

Second pass over the window wording. Tooltips, source labels, dialogs and window messages use the same DevOps terms as the buttons: public and private key, CLIENTS, golden copy, Export Keys, R2 bucket, jump host and ProxyJump, indicator, rclone.conf, public links. The sources above the SSH KEYS panels are named GOLDEN COPY and EXPORT KEYS.

The English window no longer shows Russian tooltips. 25 markup tooltips and the F3 ССЫЛКА button had no English pair: routes, SSHD, DNS, clouds, the BASE router tunnel wizard, RDP, ROUTER WEB, SOCKS, tick by mask. They have pairs now, as do the LAPTOP and SECOND VPS source tooltips, the R2 CREDS dialog title and the cloud golden copy runs. Pairs that had drifted from their text (SSHD, BASE, routes) caught up; 40 pairs with old words that nothing called any more were removed from the dictionary.

The SSHD tooltip says the key file is brought in line with CLIENTS - which is how it has worked since 1.10.8. **The C MAIN VPS tooltip** no longer promises that import-keys.cmd is enough on a new machine: EXPORT KEYS does not collect the cloud password `Tunnel\clouds.key`, it is carried by hand.

A new test. `Tests\Test-Window.ps1` checks that every Russian string in the window and dialog markup has a dictionary pair.

1.11.4 — 11 September 2026

SSHD, NDMS API, ДИАГНОСТИКА and ROUTER WG PEER start from the window again. The window hands every wrapper the data folder with `-Root`, but `Setup-SshServer.ps1`, `Check-Router.ps1`, `Check-Routes.ps1` and `New-RouterPeer.ps1` did not know that parameter and failed at binding: "A parameter cannot be found that matches parameter name 'Root'". They accept it now; `SSHD` passes the folder on to its elevated window, and `ROUTER WG PEER` takes its data from it. In `Setup-SshServer.ps1` the `$Root` variable named the script folder and would have overwritten the parameter, so SSHD would look for CLIENTS in the wrong place; it is renamed to `$Here`.

A window-to-console match test. `Tests\Test-Cli.ps1` collects the switches of every script call in the window - literal, from variables, from wrappers and `-Root` - and checks them against the script's `param()`; it also checks that `-Root` is not overwritten in the script body.

1.11.3 — 11 September 2026

1.11.3 — 11 September 2026

The TUNNEL and REMOTE rows fit on one line. Button side padding is 6 px instead of 8, the gap between groups 10 px instead of 14. At a window width of 1520 (a 1920 screen at 125% scaling) both rows fit on one line; words and captions are unchanged.

1.11.2 — 11 September 2026

NET and TOOLS. The СЕТЬ and ИНСТРУМЕНТЫ group labels became `NET` and `TOOLS` - in the TUNNEL row and in SETUP: the TUNNEL row did not fit.

1.11.1 — 11 September 2026

Group labels are one word. ЭТА МАШИНА became `LOCAL`: in the TUNNEL and SSH KEYS rows, above the file panels instead of "this PC", and in dialogs. ЧЕМ ОТКРЫВАТЬ became `ОТКРЫТИЕ`, ОТКРЫТЬ ПАПКУ became `ПАПКА`. In a 1920-wide window the long labels pushed buttons onto a second line.

The REMOTE row wraps whole groups, like TUNNEL and SSH KEYS: SOCKS no longer drops alone under the routers.

1.11.0 — 11 September 2026

Buttons carry DevOps names, with captions underneath. A button is named after the program, command or file behind it: `SSH-KEYGEN`, `SSH-COPY-ID`, `SSH-KEYSCAN`, `IMPORT KEYS`, `EXPORT KEYS`, `R2 CREDS`, `CLOUDFLARED`, `RCLONE.CONF`, `PORT 22 TCP PING`, `ROUTER PASS`, `NDMS API`, `ROUTER WG PEER`, `RDP`, `ROUTER WEB`, `SOCKS`, `SYNC ALL`, `SYNC SELECTED`, `DRY RUN`, `RELEASE`, `PRIVATE BACKUP`, `F11 COPY →`, `F12 SYNC →`, `BISYNC ⇄`. Groups are `AUTHORIZED_KEYS`, `EXTERNAL`, `МАРШРУТ`; the `TRANSPORT` section became `REMOTE`. Under the buttons sits a two-or-three-word caption in the window language that shows the direction: "keys → Tunnel\Export", "R2 → six folders". Buttons are 6 px taller; the machine buttons above the file panels have no captions. The TUNNEL row now wraps whole groups, like SSH KEYS: in a narrow window NETWORK and TOOLS take the first line and THIS MACHINE the second - the middle of the row used to fold into a column of single buttons.

REVERSE SSH TUN. Formerly SSH BEACON. The button caption shows the VPS the channel holds through right now: `REVERSE SSH TUN · SECOND VPS` or `· MAIN VPS`. The wrappers are named after the VPS: `reverse-ssh-second-vps.cmd`, `reverse-ssh-main-vps.cmd`, `reverse-ssh-check.cmd`, `reverse-ssh-remove.cmd`.

REMOVE TASK is one button. The old Task Scheduler task is deleted by the button in SETUP, which shows only on a machine where the task remains. The ОТМЕНИТЬ button in STORAGE is gone.

The window test checks that every caption under a button has a translation; the tunnel test checks that the VPS in the caption comes from the host name.

1.10.8 — 11 September 2026

`SSHD` **keeps the key file from falling behind.** The file `C:\ProgramData\ssh\administrators_authorized_keys` used to be appended to only: a machine key replaced in the store stayed admitted, and the new one did not arrive until the button was pressed again. `setup-ssh-server.cmd` now replaces the line of a machine from `Tunnel\SSH\clients` when it carries another key, adds the missing halves and collapses repeats. Lines of machines the store does not know stay and are named by their label: `РАЗДАТЬ КЛЮЧИ` on the neighbours brings them. An empty store leaves the file alone. The file is read and written as UTF-8 without a BOM. Covered by the new `Tests\Test-SshServerKeys.ps1`.

1.10.7 — 11 September 2026

The router password lives in the data folder. The working `router_*.secret` file sits in `Tunnel\RemoteDesktop\config` of the data folder, not next to the scripts in the program: it is this machine's secret, and the program folder travels to other machines as a copy. `NDMS`, `РОУТЕР`, `КОНФИГ` and `check-routes.cmd` read it from there. A file left in the program folder is moved on first use if the data folder has none yet. The password is no longer part of the machine state.

1.10.6 — 11 September 2026

Where Windows keeps machine keys. The `SSHD` tooltip, the user guide, SKILL.md and the prompts name the file: `C:\ProgramData\ssh\administrators_authorized_keys`. For an account in the Administrators group Windows sshd reads keys only there and ignores `.ssh\authorized_keys`. The program already wrote there; the texts only said "authorized_keys".

1.10.5 — 11 September 2026

The machine fetches the backup. The tooltip of `СНЯТЬ РАСПИСАНИЕ` and SKILL.md said that MAIN VPS comes for the data itself. It does not: MAIN VPS puts the nightly backup into R2, and each machine fetches it on its own — with `ОБНОВИТЬ ВСЁ` or `sync-backup.cmd`, the way the laptop does. MAIN VPS never comes to a machine, and BASE is no exception.

1.10.4 — 11 September 2026

Mirror folders without provider names. MAIN VPS units, scripts and documents live in `MAIN VPS Backup`, the SECOND VPS legs in `SECOND VPS Tunnel Config`: that is how `config\mirror.sample.json` and the user guide name them. A provider name in a folder name goes stale with the first move. On a machine with an older mirror plan, rename the folders together with the `folder` entries in `config\mirror.json`; otherwise the next update starts them anew next to the old ones.

1.10.3 — 10 September 2026

Only MAIN VPS and SECOND VPS. In every language and every text: the window, the dictionary, tooltips, scripts, samples and documentation. Descriptive names are replaced with the names. Buttons and captions: routes `SSH MAIN VPS` and `SSH MAIN VPS CF`, source `MAIN VPS`, panel captions `BASE · MAIN VPS` and `BASE · MAIN VPS CF`; in `CLOUDS` — `ЧЕРЕЗ MAIN VPS`, `НА MAIN VPS` and `С MAIN VPS` (in English `VIA MAIN VPS`, `TO MAIN VPS`, `FROM MAIN VPS`).

Routes. The choice of path to a machine is called a route everywhere: captions, tooltips, the window log, `check-routes.cmd` and documentation.

Routers have a WG server and a WG client. Texts describe routers in no other words.

The beacon is SSH BEACON everywhere. Leftovers of its former name in the `ssh-beacon-*` wrapper description and in comments are replaced.

Terms were replaced in 59 files of the program and documentation. Older changelog entries are not rewritten: they are history.

1.10.2 — 10 September 2026

No "WG SECOND VPS" tunnel in the texts either. The machine WG tunnel to the second server was removed from the program on 09.09.2026, but leftovers stayed, and the depersonalisation gave them a

new name — "WG SECOND VPS". Now gone: the mention of that tunnel in the WireGuard install button tooltip, the WORK panel tooltip about its tunnel address, the "LAPTOP only through the collection point" tooltips (nobody connects to the laptop from afar), 30 translations of the removed wizard and old tooltips, an unused liveness check of that tunnel, the filtering of `10.13.*` addresses in the router tunnel wizard, and the "the tunnel is the road" note in the transport template.

Firewall rule without the retired subnet. `setup-ssh-server.cmd` no longer admits the `10.13.0.0/24` subnet to port 22. The rule is recreated on every run, so on a machine it is updated the next time file receiving is set up.

`check-routes.cmd` **checks the machine's only tunnel.** The tunnel section looked for the `home` and `laptop` services, and only under WireGuard. Now it is one tunnel — WG BASE-ROUTER — under any of its names (`router-base-wg`, `base`, `home`), under WireGuard or AmneziaWG, and the handshake is asked from whichever client answers. The diagnostics button tooltip and SKILL.md say the same.

1.10.1 — 10 September 2026

A second review of 1.10.0 confirmed the four regressions closed and found five more places; all were confirmed and fixed.

The SSH BEACON key has no passphrase under PowerShell 7 either. The beacon script created its key with its own `ssh-keygen -N '""'` call, and the wrappers prefer `pwsh`, where the two quotes become a passphrase. The key was created with exit code zero, and the holder with `BatchMode` could not open it. Key pairs are now created by one function — `New-AbkKeyPair` in the new `Engine\KeyPair.ps1`, called by both the window and the beacon — and the pair is checked at once. If the beacon key is already locked with a passphrase, setting up says so and stops: keys are the owner's to change.

Beacon removal and inspection go to the beacon's own server. Both used the default legs to MAIN VPS. A beacon set up through SECOND VPS was removed "successfully": its key was not on MAIN VPS, that server answered "ok", and the key stayed on SECOND VPS. The legs now come from the beacon's state, the state remembers where the key was registered, every leg is asked separately, and "no key here" from another server does not count as success. If a registration place does not answer, removal fails and keeps the state for a retry.

The SECOND VPS port reaches the connection. `ADDRESSES` wrote the port, but the resolver never returned it: the tunnel client wrote the block without `Port`, the window probe and `check-routes`

knocked on 22, and files and desktop through the second server went without `-p`. Everyone uses the port now, and moving the entry from `~\.ssh\config` takes it as well.

The second server's names in `ADDRESSES`. Three fields below the table: the host name in `~\.ssh\config`, the name over cloudflared and the key name. Names with spaces or symbols are not written.

Include in ssh config. The name check looked for a `Host` line in the main file only, so a `TRANSPORT` rebuild replaced the second server's working section with a placeholder when the host was described in an included file. The parser now follows `Include`, the check also asks `ssh -G` itself, the two names are checked separately, and a placeholder with nothing to fill it no longer overwrites a working line: that line is taken from the previous transport, and the window says what it kept.

The SSH BEACON dot counts only its own channel. The indicator accepted any ssh with the same port number. The channel now is the ssh started by this machine's holder, with its key and its port, alive for more than twenty seconds; a younger one is "coming up".

Checks: the new `Test-Beacon` — a pair without a passphrase and a locked one recognised, stopping the channel, choosing legs, the removal verdict, the own indicator; `Test-Targets` — Include, the port, the names; `Test-Transport` — working lines survive a rebuild.

1.10.0 — 10 September 2026

The second server moves over by itself. A machine set up before the `second` entry keeps its second server only in `~\.ssh\config` — that is how the tunnel client wrote it. 1.9.0 did not recognise such a machine: it handed out the sample names, and the very first `TRANSPORT` build pointed the second server's panel at `second-vps`, a host that exists nowhere, while reporting "built". Now the window finds the second server in the tunnel client's block at start — the host that is not the main one, and its cloudflared variant beside it — and moves the entry into `targets.json`. A machine without a second server gets an empty entry from the sample. The `TRANSPORT` wizard fills in the second server's name only when `~\.ssh\config` describes it, and otherwise leaves the field empty and says what to do.

The second server's address — with `ADDRESSES`. A `SECOND VPS` row stands in the table next to the machines and writes the `host` field of `targets.json`. After writing, the window checks `~\.ssh\config`: no name — it points to `INSTALL` in `TUNNEL`; a different address — it says where to fix it.

The `SSH BEACON` button. The reverse ssh forward — the main way to call a machine from outside — lived in the console only. The button in the "this machine" row of `TUNNEL` raises it through SECOND

VPS or through MAIN VPS only, inspects it and removes it; every action is the existing `ssh-beacon-*.cmd` wrapper. The dot is green while ssh holds the forward. The script's texts call the forward `SSH BEACON` again, like the button and the wrappers: `SSH SERVER` stays the name of a route.

Removing the beacon no longer leaves ssh behind. In PowerShell 5.1 a single channel holder came out of its function as a lone record without `Count`, and the check skipped the search for its ssh: the holder stopped, ssh kept the port on the server, and the script reported "channel down". The holder list is now always an array, removal waits for both the holders and their ssh, and a refusal to stop ssh makes the removal fail. Setting up no longer raises a second holder beside an old one that did not stop.

The checks tell the truth. `Test-Cli` compared a wrapper's full path with the mask `*\Tools\*`, while the project lives in `E:\TOOLS`: every wrapper was skipped, and "every wrapper has a button" shone green without checking one. The path is relative now, the number of checked wrappers is printed, and the check knows the window also calls scripts directly. `Test-Targets` copied the working `targets.json`, which the sources do not carry, and failed on its first line on a clean copy; it now starts from the sample and checks the second server's move on a synthetic ssh config.

Documents. README describes the machine profile as it is — the data folder itself — and no longer asks for a `SCHEDULE` step: the program registers no tasks. SKILL.md and the user guides no longer describe the WG beacon and its key, and the `REMOVE` tooltip no longer points to a button that is gone.

1.9.0 — 10 September 2026

The second server comes from `targets.json`. The collection point's address, its host names in `~\.ssh\config`, its key and its account are no longer written into the code. The window, the tunnel client, desktop, files, the route check, the beacon and the key import take them from the `second` entry of `targets.json`, through the shared `Engine\Targets.ps1`. The sample carries neutral `second-vps`, `second-vps-cf` and `second_vps_ed25519` and no address: without one the point is not probed, and the tunnel client leaves the second server's existing entry in `~\.ssh\config` as it was and says so.

Transport sections `second` and `second-cf`. They replaced the old names in the template, and the `TRANSPORT` wizard fills in the host names. On a machine whose transport was built by an earlier version, the second server's panel comes back after rebuilding it with `TRANSPORT`.

Servers in the texts are MAIN VPS and SECOND VPS. Captions, tooltips and documents no longer name the providers, nor the second server's city and country. The Cloudflare tunnel identifier left the

tunnel client's hint.

Everyone has their own mirror plan. `config\mirror.json` left the history and `config\mirror.sample.json` ships instead: the mirror takes the sample until a list of its own exists. The plan itself became a machine difference — it is kept in the data folder and comes back from there into a fresh unpack.

Places and addresses come from the data too. The texts name places by their machines — BASE and HOME — instead of household words. The direct desktop wrappers `connect-*-lan.cmd`, the route check and the tunnel wizard take the desktop's and the router's addresses from `targets.json` rather than from the code; the tunnel's default network is a single `192.168.1.0/24`, and real ones live in `routes.txt`. The setup prompts for the fleet moved from the program to the data folder. `install-wireguard.cmd` works again: it called a script that no longer existed. The remote access README describes what exists now and says plainly that the RDP desktop has not been tried yet.

Checks: `Test-Targets` — the sample has the second server without an address and with neutral names; `Test-Transport` — the `second` and `second-cf` sections are built; `Test-Machine` — six machine differences.

1.8.0 — 10 September 2026

Window settings are part of the program again. `config\gui-settings.json` — theme, language, terminal, archive root, `СИНХРО` pairs, chosen routes — used to count as a machine difference: after every write the window copied it into the data folder and brought it back from there into a fresh unpack. Now the file lives in the program folder only. A build unpacked without it starts with the default settings.

The path to the data folder is no longer written into the settings. The `DataRoot` field repeated the `config\profile.json` pointer, and the window only ever wrote it: the data path comes from the pointer or from the search beside the program. The copy went stale on the first move — in the laptop kit it pointed at `E:\TOOLS\Audion Backup` long after the folder had become `Audion Backup Laptop`. The data path now has one owner.

`Tests\Test-Machine.ps1` checks both sides: there are five machine differences, window settings do not reach the data, and a copy left in the data by earlier versions is not brought back into the program.

1.7.0 — 8 September 2026

The program is renamed: `Audion Sync Tools`. The old name described what it started as — a snapshot of the Build stack. Today it is a two-panel dispatcher for the whole estate: machines, routers, servers, R2 buckets, clouds, terminals, tunnels, forwards, mirrors; backup is one of its ten jobs. The name joins the family — `Disk Tools`, `DevOps Tools`, `Get Tools` — and both words are plain.

The folder is now `E:\TOOLS\Audion Sync Tools`, and the mark is an "S" of two opposing arrows (`Assets\SyncTools.ico`, six sizes from 16 to 256). The name was replaced everywhere inside, transport paths included; verified after the move — both the machines and the cloud still read.

Your own application key is a separate action now. `clouds-key.cmd` replaces `client_id` and `client_secret` on a connected cloud without touching its name or anything else, and `clouds-reconnect.cmd` refreshes the token — the pass rclone travels on instead of a person. Re-creating a cloud just to change a key is no longer necessary; in the window it is the `SIGN IN AGAIN` button in the `CLOUDS` bar.

The `client_id` tail is checked immediately: it must end with `.apps.googleusercontent.com`. A key truncated on paste looks fine but makes Google answer "client not found", and the hunt costs an hour — the script now says so before writing it.

Copy and move ask first. On your own disk that is a formality, but here the other end is someone else's machine, a server or a cloud: a mistyped key costs dearly and there is nothing to undo it with. F5 and F6 now show what is taken, from where and to where, how many objects and the first names; for a move they add that the source will be left without them; if the target already holds matching names they are listed with a warning about overwriting; and when both ends are the same cloud, the dialog says the provider will move the files itself, bypassing this machine.

The other outward-writing actions ask too. The Build-stack snapshot to the bucket — with a reminder that older snapshots are rotated away and a pointer to `PLAN`. Refreshing every mirror — with a reminder that the local copy is levelled to the server. Delete, mirror and sync already asked; rename and new folder ask for a name, which is enough.

Cloud keys now survive a Windows reinstall. DPAPI seals to this user and this system: reinstall Windows and you sign in to every cloud again. The tokens themselves are not tied to the machine — only the lock is. So the reference config now lives where nothing gets reinstalled: `clouds-push.cmd` puts it on the server at `~/.audion/clouds.conf`, mode 600, and `clouds-pull.cmd` fetches it onto any machine. The password never goes to the server: it goes into the key store in the data folder and travels with the kit (`export-keys.cmd` / `import-keys.cmd`). The server holds the lock, the store

holds the key; neither opens anything alone. On a fresh machine: import the keys, then `clouds-pull.cmd` — and not a single sign-in is repeated.

A copy between two accounts of the same cloud happens inside the provider. When both ends are the same type (two Google Drives, two Dropboxes) the window adds `--server-side-across-configs`, and the data never passes through the machine: the provider moves the files on its own side. Across different clouds no such protocol exists, so the flag is added only for a matching type — the command must honestly describe what will happen.

Moving between different clouds can be handed to the server. `VIA SERVER` in the `CLOUDS` section (and `clouds-send.cmd` from a console) takes the left panel as the source, the right as the target, and runs rclone on the server: its link is a data-centre one, not a home one. The keys travel there as variables inside a script piped through `ssh … bash -s` — nothing is written to the server's disk, and only `bash` shows up in the process list. Verified: after a run, `~/.config/rclone` on the server is still empty.

The cloud check counts space and sums it up. `clouds-check.cmd` and the `CHECK` button now say, for each cloud, not only that it answers but how much is free there, and at the end how much is free across all of them. Where a cloud has no notion of a quota, nothing is written.

The wizard asks for your own application key. Without one rclone uses the shared key — the same one all of its users share: one quota for everybody, lower speed, refusals at peak hours. Google Drive, Dropbox, OneDrive, Yandex Disk, pCloud and Box accept your own; MEGA and Proton Drive offer none, they sign in with a login and password. It is optional: skip the question and the shared key still works.

Clouds work with every button, not just copy. rclone can describe a remote through environment variables, and that solved the whole thing: the config stays the ordinary one for every call, machines and buckets included, with the clouds added on top. So they get everything the machines get — F5, F6, F8, F2, F7, F11 refresh, F12 mirror, sync, the free space under the panel and marking by mask. The operations themselves know nothing about clouds, and should not. The keys never reach a command line or a temporary file: only the rclone started from here sees them.

A CLOUDS section appeared. Clouds stand as panel sources next to the machines: Google Drive, Dropbox, OneDrive, Yandex Disk, pCloud, Box, MEGA, Proton Drive — whichever you connect. A file goes from cloud to cloud or to a machine by ordinary copying, with the same buttons as everything else; there is no special transfer mode.

Cloud keys live apart and encrypted. The shared `rclone.conf` is no place for them: it lies open and travels between machines inside the key kit. Clouds live in their own `config\rclone\clouds.conf`, encrypted by rclone itself, and its password sits in `clouds.key`,

sealed by Windows to this user and this machine — a copied file will not open. The password is never displayed and never reaches the command line: rclone gets it from the environment for the duration of a call.

Signing in is done by the person, not by the program. The `CONNECT` button opens an rclone console: where the cloud offers OAuth a browser opens, where it does not rclone asks for the login and password itself. The program takes no part in that conversation — all it gets is the issued access key. The same from a console: `clouds-add.cmd`, `clouds-remove.cmd`, `clouds-check.cmd`, `clouds-keys.cmd`, and `clouds.cmd` for the list.

Marking by mask. A funnel button now sits in each panel header: a mask like `*.zip` or `2026-*`, several separated by spaces, an empty line clears every mark. The header buttons are a quarter smaller too — they are auxiliary.

The top row is split into three blocks. `ROUTE`, `MODES` and `STREAMS` each have their own frame: on a shared backing the sections read like ordinary control buttons. The route block wraps its buttons instead of cutting them off on a narrow window.

The `HOME` button became `LAN`. It is neither a tunnel nor a machine but a way to go by the local address with no middleman — and the word says so.

The route board is laid out in columns. Every road has its own column: the same road sits in the same place for every machine, and an empty cell means the machine has no such road. The row used to grow freely, and the laptop's fifth road was cut off by the window edge.

The free space of the open folder is shown under the panel. On servers that is the number that matters: will the thing you are about to copy fit. It is counted for the open folder, not for the machine as a whole — a machine with `C:` and `S:` has two different answers, and a single one would be a lie. In the root, where the list of drives stands, nothing is written: every drive has its own. The ssh answer takes a second or three, so it is asked by a separate process and remembered for five minutes — the window never waits. R2 buckets show no free space: there it does not run out, there the count is for what is stored.

Reachability dots are gone from the file panels. A source you cannot reach right now is still visible and still opens on an attempt — and the attempt tells the truth better than a probe taken a minute ago. The dots stayed where they decide something: on the machines and routers in the terminal row.

Machine buttons are built from the targets file. There were eight of them, one per machine written into the markup, and a ninth would have meant edits in three places. The set now comes from `targets.json`: a key prefixed `router-` goes to the router group, the rest to the machines, in file order. The tooltip is built from the same file — the target description, the hint for when it stays silent,

the note about password-only login. The `ADDRESSES` wizard no longer keeps a machine list of its own either.

SECOND VPS is now a panel source. The collection point opens like any other source, next to the machines and the buckets, and a file lands on it the same way it lands anywhere else. From home the road there is direct, no VPN; where its ssh is cut, the cloudflared spare works. The account and the key are not copied into the rclone config: the host `second-vps` is already described in `~\.ssh\config`, and ssh takes them from there.

The ssh beacon has two legs to choose from. `ssh-beacon-fast.cmd` raises it through SECOND VPS — it answers about forty milliseconds sooner and needs no VPN from home, but it sits inside the country. `ssh-beacon-private.cmd` goes through MAIN VPS: slower to establish, but the traffic never lands on a host inside the country. Look and removal are `ssh-beacon-check.cmd` and `ssh-beacon-remove.cmd`.

Transfer streams are set in the window. rclone splits a single file into streams only from 256 MB up, while on a wide link the gain shows much earlier: over two gigabytes to BASE one stream gave 167 MB/s, eight gave 180. The threshold is lowered to sixteen megabytes, and the stream count is picked by a spinner in the mode row — 1, 2, 4, 8, 16, 32, eight by default. The choice is remembered between runs.

The window lost a floor. Rows are now split by meaning. Route and streams are settings for the whole window, so they moved next to the modes: route on the left, streams on the right. Terminals for machines and routers, forwards for the desktop, the router web page and the proxy, and the choice of terminal program are daily work, and they gathered in the new TRANSPORT section. TUNNEL keeps how the link is built: tunnel client, beacon, routes, router keys — it is set up and repaired there, and used in transport.

SERVER MIRROR and R2 BACKUP merged into STORAGE. Both sections were about the same thing — where the copies live. The panels now show nightly snapshots, the data folder, the `audion-backup`, `audion-releases` and `audion-share` buckets and the server itself at once: a file goes from a bucket to the server and back without switching sections. The R2 upload buttons moved into the same bar, on the right.

Tunnels run on whichever client the machine already has. The machines carry Red Shield VPN, and it runs on AmneziaWG. The system holds one kernel driver for all, and whoever claims it first decides for the rest: the Amnezia build speaks both obfuscation and plain WireGuard, the plain one only plain. Hence what was observed — the VPN would not connect while our tunnels stood on the classic client. Now a tunnel is installed on what is already there: not because it is better, but so as not to take away someone's own VPN. The program installs no client of its own and touches nobody else's.

The program stopped telling tales. Tunnel state was asked of `wg.exe`, while the tunnels stand on AmneziaWG — and they do not answer the plain client at all: `wg show interfaces` returned emptiness over a working tunnel. Now every client present on the machine is asked, and the answers are added together.

The hour everything was dead. Handshake age was counted through `Get-Date -UFormat %s`, which returns local time dressed as universal. The gap is exactly the time zone: a minute-old handshake showed as six hours old, and a live tunnel looked dead.

Addresses are named by whoever hands them out. The beacon window showed taken addresses and offered a free one, counting them from its own files — from targets and from configs lying nearby. What came out was invention: the list held addresses the server does not have at all, and the one called free had long been taken. Now the window asks the collection point, and the peer wizard asks the router; if they stay silent, we say so instead of substituting a guess. The address field is gone: it is assigned by whoever hands it out.

Its own config is looked for before the issued ones. Finding none of its own, the script took the freshest issued by the wizard — whosever it was. That is how the work machine's config landed on the home one: with its key and its address, which the router does not allow that key. Packets were dropped silently, on any client, and it looked like "raised it and nothing happened".

The data folder is named after the machine. `Audion Backup Laptop`, `Audion Backup Work` — the name is visible from outside, and the `Machines\<name>` layer inside is gone: the folder is its own profile. A profile from the older schemes is lifted into the root the first time the folder is opened. On the first run the folder is pointed at by a person — it holds keys, addresses and the router password, and the program will not settle into it on a guess, not even when there is only one folder beside it.

The `RAISE ALWAYS` button. The client sets the service up for manual start, and for a machine travelling through other people's networks that is right: you switch the computer on at work and the tunnel does not disturb the VPN there until it is wanted. But some machines are out of arm's reach: the HOME machine reboots and there is no link with it until the next visit. For those, the second raise action switches the service to automatic start. Plain `RAISE` returns it to manual — the mode toggles with the same press, without removing the tunnel.

The beacon takes its name from the machine and its address from the server. The parameters held the name `laptop` and the address `10.13.0.2`, and the second machine was supposed to type in its own by hand. It was forgotten: machines were issued under one name and took each other's address, and after a rename the server kept a peer under the old one. Now the name comes from the machine, and its own peer is recognised by key and keeps its former place — the key survives both a system reinstall and a rename.

Removing a peer removes all of it. One line with the key was cut from the server config, while `[Peer]`, the name and `AllowedIPs` stayed behind. A peer without a key does not work, yet it holds an address — and the next install took the address after it. Two rounds of "set up, removed" walked the beacon from `.2` to `.5`, and stumps piled up on the server that the working tunnel knows nothing about.

The client is taken in the machine's own architecture. Red Shield ships amd64 and arm64 builds side by side, and launching the wrong one is no quiet error: Windows shows a modal "machine type mismatch" box that neither `try` nor `catch` will catch. State is polled on a timer, so the box kept surfacing over the work again and again.

The inventory shows a matrix. Who goes where, as a table: keys down the side, machines across the top, a plus, a dot, or a question mark where the machine did not answer. Comma-separated lines showed who was registered where, but not where the holes were.

Router buttons ask which router. There will be more than one: BASE has its own, HOME will get one, WORK later. Only routers with an address are listed — while there is one, no question is asked.

Only the `WIREGUARD` button installs a client, and it looks first. The beacon script, finding no WireGuard, pulled it in through winget by itself — onto a machine that had carried AmneziaWG for months and had no use for the plain one. Now the button says which client is there and why a second is not needed, and installs only when there is none at all.

Wrappers moved into `Runs`. The program root is clear: documents, folders and `Start.exe`.

The setup row is laid out across the width. `ШАГИ` on the left, `ИНСТРУМЕНТЫ` in the middle, `ЗАДАНИЯ` on the right — in one line. As two stripes the row read worse: the upper one hugged the left edge, the lower one huddled at the right, and the middle stood empty. The `ПАРК` group had emptied earlier — its buttons moved to the keys section — and is gone along with the rule that separated them.

Button captions no longer look cut off. A trailing ellipsis — `ТЕРМИНАЛ…`, `АДРЕСА…`, `КОНФИГ…` — meant "opens a chooser", but read as text that ran out of room. What a button will ask is said in its tooltip, which has a whole line for it.

The terminal caption is translated as a whole. It is assembled in code, and only the value at the end went into English: the result was a mix — `ТЕРМИНАЛ: own`. Now the whole string is translated, and the caption is rebuilt when the language changes: the tree walk used to put the markup's own word back in its place, losing the chosen program.

The tunnel row is in one line as well. `СЕТЬ` on the left, `ИНСТРУМЕНТЫ` in the middle, `ЭТА МАШИНА` on the right. The tools stood on a second tier and read as a row of their own. The middle is a wrapping panel: from about 1400 points of window width everything sits in one line, and if the window is narrowed further the buttons drop down instead of being cut off at the edge.

The window language is taken from the settings from the start. The code had a hard `en`, so with a Russian window the language variable stayed English: the markup was restored by the tree walk, while everything assembled in code — captions and log lines — went through the dictionary and came out in English. On the button it looked like `TERMINAL: own`.

Removing the beacon does not ask for rights when there is nothing to remove. The service is already gone — what is left is the peer at the collection point, and that is an ssh conversation. The rights are worse than useless here: the elevated process never reached the server, though the key works from an ordinary window.

An ssh refusal names its reason. What ssh said on the error stream was thrown away, leaving "check the SECOND VPS key" — guess whether it is the file permissions, the wrong key, or a silent server. The reason is now printed in words: `connect to host ... port 22: Connection timed out` — plainly the network, not the key.

The file panels' background got lighter — `26 33 38` instead of nearly black. Frames, the ribbon, the terminal block and the footer are untouched.

The `DNS` button: who resolves names, and who to trust with them. This is what a break that isn't there looks like: addresses ping, the route is in place, and not a single site opens. The answer is usually the same — the adapters point at a local resolver that is not running: `ControlD`, `YogaDNS`, `AdGuard` and `dnscrypt` put themselves on `127.0.0.1` and resolve names over their own encrypted channel.

The button first shows what it found: the addresses on the adapters, what the router offers over DHCP (Windows keeps it in the registry next to the manual one, and it does not go away), who holds port 53, and who ends up asking outward — the last hop is found through Google's service name, and the reverse record names its owner: `dns.google`, `one.one.one.one`, a node of the provider.

Then it offers a choice: leave as is, back to the router, or one of Cloudflare, Google, Quad9, Yandex, AdGuard. The chosen one is asked for a name first and written only if it answered: a silent resolver would leave the machine without names, and the repair would be blind. A local resolver's service is started if it is down. Tunnel adapters are left alone.

The same button checks the `NlaSvc` service — the one that decides whether there is a way out and draws the tray icon. With it stopped you get "the internet is sort of there, but": sites open while

Windows shows a globe, the connection profile sits on `LocalNetwork`, and the firewall picks its profile blindly. The button puts the service back on automatic start — Windows' own default — and starts it.

The private key is closed everywhere it lies. Only the `.key` file was protected, while the same key sits as a `PrivateKey` line inside `.conf` - and configs were written with permissions inherited from the place. On `E:\TOOLS` that means Users with read and Authenticated Users with modify: a neighbour on the system could read the key and replace the config, and the replacement would take effect the next time the tunnel came up.

Permissions are now set by one routine - `Engine\Secrets.ps1`. It closes a file or a folder down to three subjects (owner, administrators, system) and verifies the result: "protected" has to be checked, not promised. The beacon installer closes the whole key store on every run - permissions travel with the place, and a moved data folder opens the store again. The router-tunnel wizard and peer issuing close their configs right after writing them.

The `-AtBoot` switch survives elevation. Name, address and removal were passed into the elevated process; automatic start was not. A run with `-AtBoot` from an ordinary console went through UAC and landed in the manual-start branch: the person asked for "come up at boot", and the machine stayed silent after a reboot.

Inspection stopped moving files. The `-Check` switch promises "look only", yet it created folders and moved the old config into the data folder, fixing its permissions along the way. If the service expected the file at the former path, its next start could fail. The migration and the folders now happen only when we are really installing.

DNS "the router itself" - one per adapter. The first gateway found was written to every adapter at once. With Ethernet and Wi-Fi on different networks one of them got the neighbour segment's gateway, and lost names as soon as that link went down. The address is now taken per interface, each gateway is checked on its own, and an adapter without a gateway of its own is left alone.

The encoding check no longer looks into `._runtime`. Unpacked tools, working copies and audit material live there - expecting BOM and CRLF from them is pointless.

A beacon without a tunnel: a reverse forward to the foreign server. The old beacon raised a WireGuard adapter and service in the system - it took over routes, set its own DNS, fought the paid VPN over the shared driver and failed at boot while the network was not ready. The new one does none of that: the machine itself holds one outgoing connection and asks the server to open a port on its localhost. Whoever reaches the server gets to the machine through that port. No adapter, no routes, no driver.

This is a second road beside the old one, not a replacement: the two beacons are installed and removed separately. WireGuard gives the whole machine - any port, files, the desktop; the forward gives port 22 and whatever ssh can carry through it. In exchange the forward is TCP, and it passes from under someone else's VPN and a mobile carrier, where UDP does not come back at all.

The legs are tried in turn: first the server's direct address, and on a break the same server over cloudflared. Nothing is asked, because the network underfoot keeps changing: what failed a minute ago may be the only one that works now. `-Check` shows the order of legs and which one holds the channel, and the holder writes `logs\ssh-beacon-<machine>.log` - it lives inside a task, without a window, and everything it said used to go nowhere.

The beacon key is separate from the machine key and is registered on the server with `restrict,port-forwarding,command="/bin/false"` - no shell can be had with it. The `authorized_keys` edit keeps a copy of the file and checks the line count: the owner's keys live there too.

The `МАЯЧОК` road in the window. A fourth path button, between `CF` and `SECOND VPS`. It leads to the same machine but does not depend on anyone reaching it from outside: the machine holds the channel itself. For a machine behind someone else's NAT it is the only road at all, so the automatic choice puts it before the collection point.

The server no longer holds a port for a dead session. `sshd` does not notice a break at once, and the forwarded port stays taken - a new connection gets `remote port forwarding failed` and the beacon never comes up. Fixed from both ends: `ClientAliveInterval 30` and `ClientAliveCountMax 3` on the server, and the holder waits a minute instead of hammering at a busy port.

The beacon no longer takes someone else's peer off the server. A block was removed on a match of any of three marks - key, name or address. The key is the peer's identity; the name and the address may belong to another machine, so an explicit `-Address` that matched a foreign peer silently deleted it. Now the own block is found by key, and a name or address match under a foreign key is a conflict: the install stops and says whose place is taken. `-Force` insists - then the overriding block goes deliberately, not behind one's back.

Address and peer write are one transaction under the server's lock. These used to be two conversations: "which address is free", then "write me down". A second installer fitted between them - both got the same address, and the second overwrote the first. Choice and write now go as one command under `flock`, and the temporary files are unique: the shared `/tmp/wg2.new` and `/tmp/wg2.strip` collided under concurrent work as well.

A server refusal stopped looking like success. The commands ended with an unconditional `echo ok`, the shell did not stop on error, and in the sync step the final code was overwritten by a successful cleanup. The install carried on and raised a service for a peer the server does not have. Every step now runs under `set -e`, the answer is checked together with the exit code, and on refusal the local service is not touched at all.

Proven on a bench, not promised. The real command pieces were cut out of the script and run against a mock config with three peers: a foreign address and a foreign name produce a conflict and touch nobody; with `-Force` the overriding block goes and no duplicate address is left; one's own key rewrites one's own block; an unknown key touches nobody.

The file client stopped trusting whoever answers. WinSCP was given `-hostkey=*`, accepting any host key. In someone else's network with the same private address (`192.168.1.x` is everywhere) the neighbour's machine answers on port 22, and the files would go there; the tunnel to the server does not help - it verifies the server, not the machine at the far end. The fingerprint now comes from our own host key list and is handed to the client; with no record the connection does not open at all, with a hint to collect the keys by button. The plain `sftp` got the same list instead of a throwaway file: `accept-new` into an empty file means "I trust whoever answered first".

Host keys are updated per node. The file used to be rewritten entirely with whatever was collected right now: if only BASE answered, the records of every other machine vanished. An unreachable machine now keeps its previous keys, HOME and WORK were added to the walk - they were missing altogether - and the port is taken into account.

A changed host key no longer passes in silence. It looks the same whether the system was reinstalled or the host was substituted - the difference is whether a person knows about it. The button keeps the old record and says which machines have a different key; the new one is accepted by the same click with `Ctrl` held.

An ambiguous data folder stops the work. Kits for all machines lie side by side, and without a pointer the first one alphabetically was taken - the tunnel installer picked keys and configs out of a foreign kit while the changes went elsewhere. The search now stops with a question, and the chosen folder is passed to the scripts by `-Root` - through elevation and into the beacon's task as well.

The previous server snapshot goes only after the new one is checked. A matching checksum says the downloaded bytes equal the source - not that the source is any good. The server could ship a truncated archive together with its correct sum, and the previous, working one is already gone by then. The new snapshot is now opened and listed first - read-only, nothing unpacked - and only then the old ones are removed. If it does not open, the previous ones stay and the window says so.

Choosing a project limits the transfer as well. `-ProjectName` filtered the list on screen while `rclone sync` was handed the whole folder: with several projects side by side, other people's archives went up too. Worse, the check then compared the full remote directory against the selected part of the local one and declared an error - after the extra files had already been sent. The transfer and the check now hold to the same scope.

The router's file wrapper works again. It called the target `router` while `targets.json` holds `router-base`, `router-home` and `router-work` - the entry ended with "target not described". The short name is kept as the familiar one: it expands to the BASE router, and the list of allowed targets now matches what actually exists.

Port forwarding: the desktop and everything else without a tunnel. The window carries files, the terminal runs commands, and a desktop, a router's web page or a network share now travel over the ssh you already have. The address is resolved on the far side, so one machine gives access to everything it can see - itself, the router, a printer, a neighbouring computer.

Nothing changes in the system: no adapter, no routes, no DNS of its own, and another VPN has nothing to argue with. Close the desktop window and the forward is gone.

Added: `Tunnel\RemoteDesktop\Open-Forward.ps1`, four wrappers and the `ДОСТУП` group in the `TUNNEL` row - `РАБОЧИЙ СТОЛ`, `ВЕБ РОУТЕРА`, `ПРОКСИ`. The last one opens the whole home network to a browser over SOCKS5.

A tunnel is still needed for exactly one case - when the network is needed whole and at once, including what was not named in advance. For a single port a forward is cheaper.

A tunnel is not raised over the addresses of the network the machine is in. It takes everything listed in its `AllowedIPs`; if such an address already belongs to the local network, it disappears for the machine - the packets go into the tunnel instead of to the neighbour. It happens in a network that takes the whole of `192.168.0.0/16`: the road into the BASE network would take addresses from it at once.

But an overlap of ranges is not yet a conflict: only what is taken can be taken away. When the addresses needed are free in the large network, the tunnel works there.

So the check goes in two steps: the arithmetic - does the address fall into the local network - and a live probe, does anyone answer on it here and now. The power button names only real clashes. At home this means "the tunnel is not needed, the addresses are right here"; in someone else's network - "it will take these addresses away". Raising it anyway is a click with `Ctrl`. Lowering is never checked: dropping a tunnel is always safe.

The network arithmetic lives in `Engine\Routes.ps1`: it reads `AllowedIPs` from the config, takes the machine's live addresses with their masks and compares by the shorter prefix.

1.6.1 — 7 September 2026

The tunnel is found under its former name too. The name changed twice: `home`, `base`, now `router-base-wg`. The wizard removes the old ones on install, but until it has been run the service lives under the old name and works — while the program looked only for the new one and reported "not registered" for a running tunnel: the dot grey, nothing to switch off. The service is now looked up under all three names, and the wizard shows in a badge which one it was found under.

A refusal of rights is recognised by deed, not by words. Windows answers an attempt to stop someone else's service with `Cannot open 'WireGuardTunnel$base' service on computer '.'` — neither "rights" nor "access" appears in that line, and a word-based check stayed silent: the dialog offering to grant the right never appeared, leaving a bare "did not work out". The test is now different: the service exists, the switch did not go through, there was no timeout — so we offer the rights.

1.6.0 — 7 September 2026

Tunnels are switched on and off by a button. Next to `ROUTER-BASE WG` and `BEACON` there is a power button: `▶` to raise, `■` to drop. The wizard stays what it was — it installs a tunnel and removes it for good; the button only raises and drops the service, leaving the config and the keys in place.

This is needed because a raised tunnel takes the listed networks for itself and gets in the way of ordinary network work: another VPN, the local network, the printers. Switching it off happens often, and before that meant opening the WireGuard application or digging into Windows network settings.

To make switching instant rather than a rights prompt on every click, installing a tunnel now grants this account the right to start and stop that one service — and nothing else: no installing, no removing, no changing settings. On tunnels set up earlier the right is granted on the first refusal, with consent and once.

1.5.1 — 7 September 2026

A chosen `Machines` no longer breeds `Machines\Machines`. In the folder picker it is easy to go one step too deep: it shows a tree, and `Machines` looks no worse than its parent. The path was then joined twice, and the program said nothing about it. Now any of those folders resolves back to the

data folder — `Machines` itself, a profile inside it, and even a nesting that has already happened; the log says what was given and what was taken.

1.5.0 — 7 September 2026

The program no longer keeps anything of its own. One file is left — `config\profile.json`, two lines: where the data folder is and what this machine's profile is called. Everything else — addresses, bucket keys, host fingerprints, the router password, window settings — lives in `Audion Backup\Machines\<Name>`. The program can be deleted, rebuilt and unpacked again: it reads the pointer and finds its own.

The profile is named by a person. The `DATA` button asks for the folder, then for the profile: a list of what already lies in `Machines`, plus a field for a new name. The name is not derived from the computer name: at work it cannot be changed, and the data folder travels between machines — a guessed name would start an empty profile there instead of the one brought along.

Addresses are filled in as one table. The wizard used to ask about one machine per click and close; nothing said the button had to be pressed four more times. Hence the outcome: one machine got an address, the transport was built with eight unfilled placeholders, and the window reported success. Now all machines at once — address, account, port — with a note on what is silent and why that is not a write failure.

The window log is written to a file — `logs\gui-YYYYMMDD.log`, one timestamped line each. Before, only crashes reached a file, and there was nothing to examine on someone else's machine: the panel shows everything, and nothing can be sent from it.

The address frame ships with the build. `targets.json` is not only data — the machines themselves are declared in it; writing an address edits the text of an existing file and silently answered "failed" when it was missing. Now `targets.sample.json` travels beside it, and the program lays it out on first run.

1.4.0 — 7 September 2026

What makes this machine different now lives in the data. Next to the program, in `Audion Backup\Machine`, sits its layer: window settings, the rclone transport, bucket keys, host fingerprints, machine addresses and the router password. On start the program takes from there whatever it lacks; on every write it puts the fresh copy back. Unpacking a new build over a deleted old one no longer carries any of that away — it used to carry away all of it, and putting it back was hand work.

The restore takes only what is missing: what a person is working with right now outweighs what was stored. Build version, themes and the config template do not go into the layer — they are the same for everyone and arrive with the program. The `LIST` button shows the whole layer: what lies where, and what is missing.

Tunnel addresses are typed into boxes. In the `ROUTER-BASE WG` wizard each address gets its own line: four boxes for the numbers, moving on after a dot or a third digit, network size as buttons in words (`ONE ADDRESS`, `254 ADDRESSES`, `65 THOUSAND`, `16 MIL`), `ONE MORE ADDRESS` adds a line, the cross removes one. The resulting record is assembled in a badge above as you type, with the machine's own network shown beside it — so it is visible whether the list touches it.

Split tunnelling is set there too. The tunnel always carried only the listed networks, but the list was baked into the code: a whole subnet overlapped the local network and took the printers with it. The list now lives in `Audion Backup\Tunnel\WG\routes.txt` and is edited by button; single hosts are written as `192.168.1.10/32`. Before raising, the script warns when the machine's own address falls into a network being routed.

Tunnels are removed by button. The `ROUTER-BASE WG` and `BEACON` wizards offer three actions: `RAISE`, `CHECK`, `REMOVE`. State is shown as badges — the service, the address inside the tunnel, the config that was found.

Fixed. State badges brought the dialog down: a local variable collided with a parameter declared with a value check. Handlers made through `GetNewClosure` did not know the window's functions and failed on click. The `CANCEL` button stayed Russian under English. Both faults are now caught by `Test-Handlers` through the syntax tree, and the layer is checked by `Test-Machine` on a sandbox.

1.3.0 — 6 September 2026

Row colours come from the Total Commander scheme in full. Colours, rule order and type masks are read from `config\colors`; the program no longer keeps an extension list of its own. The dark side takes the paired `ColorFilterNColorDark` values, the light side the plain ones. Age counts in calendar days, as in TC.

Two themes — dark and light, a button in the header. The choice is remembered; the log, the probe line and the panels are repainted with it.

Column sorting — click a header, click again to reverse. Folders stay on top, the cursor holds on to the file name.

A "modified" column — one format for every row.

Small things. PowerShell's own `Temp:` drive is gone from the drive list. Clicking a drive letter in the path bar leads to its root, not to the program folder. The rclone help text no longer floods the log.

Two backgrounds instead of four. Files, log and terminal share one tone, the blocks share it too, and the window underneath is lighter: blocks read as blocks rather than as outlines. `РЕЖИМЫ` became the same full-width block. The title bar is untouched.

A deep sea cursor. The row under the cursor is painted `#004360` with light text on it. The cursor colour no longer comes from the Total Commander theme: it is blue there and would return on every theme switch.

Marks reach the screen. Panel rows went through sorting, and `Sort-Object` wraps each of them into a `PSObject` — WPF lost the change notifications, so a tick set by the space bar or by `ВСЕ` only appeared when the list was rebuilt. The wrappers are unwrapped now. The space bar also marks the highlighted row rather than the one the keyboard cursor left behind.

A reachability dot on every source. Each source button carries a green dot when the place answers: machines go by the last probe or a live detour, folders by their presence, buckets by keys plus a way out. Unreachable ones dim. What does not fit moves to a second line instead of scrolling sideways.

Two languages. A header button switches Russian and English; English is the default. Captions, tooltips, log messages, dialogs and the probe line are all translated, and the window is not rebuilt. The dictionary lives as pairs in `Engine\Lang.psd1`: PowerShell hash keys ignore case, and the button caption would collide with the lowercase probe word. The engine scripts are translated outright — their output is read in the log anyway.

Wizards instead of a text editor. The `SETUP` section can now set up what used to be edited by hand: `ADDRESSES…` writes a target into `targets.json` and knocks on its port; `TRANSPORT` builds `rclone.conf` from the template, asking only for the Cloudflare account id and naming every placeholder it could not fill; `NEW KEY` creates the first key pair and shows its public half; `DATA FOLDER` picks the data folder, creates the skeleton and remembers it.

Two more tests. `Tests\Test-Targets.ps1` and `Tests\Test-Transport.ps1` check that a target edit touches one entry only and that the transport is built from the template correctly.

A setup section — `SETUP`. Four groups with separators: keys, machines, tools, data. Every button carries a dot: green when the thing is done, grey when it is not. The buttons call what was already written; only `АДРЕСА…`, which opens `targets.json`, is new.

Bucket keys are typed into fields. `КЛЮЧИ R2…` asks for the id and the secret and writes them into `r2.credentials` where rclone looks for them, then asks the bucket for its list. The values never reach

the log. The dialog learned several fields at once.

Dialogs follow the window theme. The palette copy in `Dialog.xaml` stays as a factory fallback; the live window hands the dialog its own brushes.

`SERVER` **among the machines.** The server sits in the access row and in the panels of `TUNNEL` and `R2 BACKUP`, so files move between the bucket and the server directly — no scp, no console. The road is rclone's own sftp; the spare one, over cloudflared, stays on external ssh. Targets gained an `sshHost` field.

Its own host key list. `config\rclone\known_hosts` knows BASE, LAPTOP and SERVER, and every panel checks against it — no more paths into the user profile. `КЛЮЧИ ХОСТОВ` refreshes it. `~\.ssh\known_hosts` is left alone.

The first test. `Tests\Test-Window.ps1` checks the window without launching it: markup and module parse, the load list is complete, every section's row and switch are in it, both themes know every brush.

The machines moved into the ribbon. `BASE`, `LAPTOP`, `HOME`, `WORK`, `ROUTER-BASE`, `ROUTER-HOME` and `ROUTER-WORK` sit in the middle of the section ribbon and are visible from every section — a console is needed outside the tunnel section too. The section button row lost its crowding and spread across the window in three groups. The three new points are prepared in advance with empty addresses: a grey dot, and pressing one says the address belongs in `targets.json`.

One button for three tools. `ОБНОВИТЬ` refreshes the tunnel client, `rclone` and the portable terminal; the separate tools button is gone. Versions are asked for before downloading, so nothing already fresh is fetched again, and `-Force` downloads anyway. Each download shows size, speed and what is left. The terminal's real release is kept in `.version`, because `wt.exe` reports the branch number instead.

The probe shows time. Milliseconds per road instead of ticks, and the window's own lag beside them — so it is clear whether the network or the program is slow. The measurement left the window thread: on it, the dispatcher queue was being measured and every road came out the same.

Panels. New sources `R2 MEDIA` and `R2 VAULT`. The panel header starts with the navigation buttons, sources follow. Four crashes fixed: a single-item listing, an empty bucket, the exception guard itself, and a status line that stayed on "reading..." forever.

Tunnels. The dot reflects a living channel, not just a running Windows service. The laptop beacon was repaired: the script still held port 8444 while the server listens on 500.

Its own name on the taskbar — the window no longer looks like PowerShell.

1.2.0 — 5 September 2026

Folder synchronisation. Three buttons to the right of the file operations:

- `F11 ОБНОВИТЬ →` — top up the opposite panel with what is missing or older; deletes nothing;
- `F12 ЗЕРКАЛО →` — make the destination an exact copy of the source; surplus files are deleted, so it asks for confirmation showing direction and scope, and a safety catch cancels the run if more than 200 files would be deleted;
- `СИНХРО ⇄` — two-way: changes travel in both directions.

Direction comes from the active panel: the source is the one holding the cursor. These buttons work not with folder names but with what is open: the two directories from the breadcrumb trails, including everything nested at any depth.

Checkboxes set the scope. Nothing marked — the whole open folder is levelled. Something marked — only the marked branches: they become `--include` filters, and `rclone` applies filters to deletion as well, so neighbouring folders in the destination stay untouched. Verified on a sandbox: a mirror over two branches out of four updated only those and left the other two alone. Unlike F5, a highlighted row does not count as scope: a mirror deletes, and the choice must be deliberate. `СИНХРО` refuses marks and says so — it remembers the state of a folder pair, and changing the scope would require a fresh baseline.

Four roads to the home machines. Panels take the first one available: at home directly, from a foreign network by a hop through the overseas server by name, then through the same server over cloudflared, and the SECOND VPS comes last. Transfer speed is the same on the far roads (1.65 MiB/s measured either way), but a session over the direct name opens twice as fast — 1.7 s against 2.8 s. There is no risk in putting it first: the probe tests that address, and once it stops answering the window moves to cloudflared, where there is nothing to block. The choice is made by the probe, which now tests the server as well as the home addresses; its result sits in the mode strip instead of the button row, where it used to overlap the outermost buttons.

РОУТЕР and КОНФИГ buttons. Talking to the home router over its own API is now in the window too: `РОУТЕР` shows model, firmware and peers with their handshakes, `КОНФИГ…` issues a WireGuard config for this machine — it asks for a name in its own dialog and first shows the commands that would go to the router.

Empty folders now travel. For `rclone` only files exist: a directory with no files in it is never created on the other side and is silently left out — a mirror finished in a second with `0 B / 0 B` while folders were missing in the destination. Every transfer, the F5/F6 queue included, now runs with `--create-`

`empty-src-dirs`. Empty, hidden and system files travelled before as well — but the "hidden" and "system" attributes are not preserved, and the copy becomes an ordinary file.

File timestamps travel. Every transfer also carries `--metadata`, so a copy remembers when the file was created, modified and last opened. Without it only the modification time survived and the copy claimed to be created today. One caveat concerns machines over ssh: the `rclone` sftp backend supports no metadata at all, so only the modification time arrives there. Timestamps are preserved in full between disks attached to this machine.

Directory timestamps are no longer set on machines over ssh. `rclone` updates a folder's time after copying, and Windows OpenSSH refuses — apparently while files are still being written into it. `rclone` treats that as an I/O error and then declines to delete anything: the mirror carried the files across but stayed unfinished and ended with exit code 1. When either side is a machine over ssh, `--no-update-dir-modtime` is now passed. File timestamps travel as before; a folder's time cannot be set on that side anyway.

Retries now pause. `rclone` retries a failed operation three times but fits all three into one second — while the usual cause is temporary: files just copied are still held by the indexer or antivirus, and a folder that has only just been emptied refuses to go with "permission denied". Retries are now five seconds apart, and deletion on a machine over ssh gets six attempts — together about half a minute for the folder to be released.

Configs issue themselves. A WireGuard client used to be added by hand in the router's web interface and its file carried over. `new-router-peer.cmd` now does it: the key pair is born on the machine (the router only ever sees the public half), a peer with a free address from the router's own subnet is created, the server's public key, port and domain name are read back, and a ready `.conf` is assembled from all of it. The same wrapper removes a peer with `-Remove` and shows the commands without sending them with `-WhatIf`.

`RouterNdm.psm1` does the talking: NDMS greets with a refusal carrying realm and challenge, and the password never travels — a digest is compared instead. It can be stored with `-Remember`, encrypted by Windows under this account on this machine, and is not asked for again.

A WIREGUARD button. Installs WireGuard itself — the program without which neither the way home nor the beacon can be raised. It used to be installed as a side effect of the first run of the home tunnel; now it is a separate action with its own wrapper `install-wireguard.cmd` and its own dot: green when the program is in place, red when it is not. WireGuard has no portable build: a tunnel is held by a Windows service and its adapter is a driver, hence an installer and administrator rights.

The button row sits on the panel splitter. It used to hold the centre of the window, and the status line ran over it: during a transfer the percentage and speed were drawn on top of F11 and F12. The row's middle now matches the splitter between the file panels and is recalculated when the splitter is dragged, while the status has a column of its own on the right and is trimmed with an ellipsis when space runs short.

The active panel is now visible. Direction comes from the panel holding the cursor, and there was no way to tell which one that was. Its frame is now highlighted in dark sea blue, and a panel becomes active on any touch: the list, a source button, the breadcrumb trail or empty space inside it.

Dialogs are ours now. A system `MessageBox` is drawn by Windows and never goes dark: a white box flashed in the middle of a dark window. Confirmations and name input now use our own window in the program's colours, and the VisualBasic `InputBox` — which looked foreign altogether — is gone. The window title bar went dark too: it is drawn by the system, and a DWM attribute asks for it.

The panel is no longer rebuilt for nothing. The two-second refresh replaced the whole list: checkboxes, cursor and scroll position were lost, and copying then complained that nothing was selected. The panel now compares the listing with the previous one and leaves the list alone while nothing has changed; when something has, the marks are carried over by name.

The wait cursor is now shown for remote sources only. Reading a local directory takes a fraction of a millisecond, yet the hourglass flickered on every automatic refresh and looked like heavy work.

1.1.0 — 5 September 2026

The window became a file manager. Two panels are always visible, and the mode strip changes only what they show and which row of buttons appears. A button stops being an incantation: pressing "lay out the keys" you see the reference on the left, the profile on the right, and what changed after the run.

One transport for everything. A local disk, a machine over ssh and an R2 bucket are read by the same `rclone`, so copying works between any two sides. Measured live: 2 GB to the workstation in 11.6 s (176 MB/s), back in 10.1 s (203 MB/s) — about two thirds of what SMB gives, in exchange for resume, retry and verification. The traffic goes through the machine running the window; at home that is the home network through the router, with no SECOND VPS involved.

A queue. F5 and F6 put the marked items into a queue instead of starting an operation at once. The destination is remembered when adding, so items can be collected from different folders and different sources and still travel to one place. Tasks run one at a time: parallel transfers would share the channel and mix their output.

File-manager keys: Space marks a row and moves the cursor down, Enter opens a folder, Backspace goes up, F2/F5/F6/F7/F8 rename, copy, move, create and delete.

You can see what happens. A progress bar at the bottom, the percentage in the window title and the taskbar, speed and remaining time in the status line, an `rclone` line in the terminal every five seconds. Panels refresh themselves: local ones every two seconds, remote ones after an operation, by button and on window focus.

Orphans no longer outlive the window. An `rclone` left from a previous run kept downloading silently — a folder deleted by hand reappeared. Now the processes of this copy are killed when the window closes and by the STOP button, and the window warns at start if such a process is already running.

Row colours come from a Total Commander theme: the values are read from `config\colors\AudionDark.ini` (decimal BGR there), while the extensions are listed in the program — the theme holds none, only references to TC search templates. Folders are yellow, as in Explorer.

Small things fixed along the way. A drive root must end with a slash: "E:" means "the current folder on that drive" to Windows, so entering a disk led into the program's own folder. For ssh the root "/" is not the same as an empty path: the first lists the machine's drives, the second its home folder. `rclone` prints progress only with `--stats-log-level NOTICE`. The scrollbars and the progress bar are drawn by hand — the system ones look like a patch in a dark window.

1.0.0 — 5 September 2026

The first build. The project is assembled from what had spread over two other folders during the previous year: the mirror scripts lay mixed in with the mirrors themselves, the snapshot of the private archives was taken by the publisher, and the mirror borrowed the bucket keys from its neighbour.

The program is separated from the data. `Audion Sync Tools` holds the window, the scripts, the tunnel client, the access to the machines, its own `rclone` and its own keys to the bucket. The neighbouring `Audion Backup` stays a store: snapshots, six mirrors, the key reference, the tokens — and not one executable file. The path to the data is worked out from where the program stands and can be given by `-Root`; no drive letter is hard-coded, so on a machine where the tools live on another disk nothing needs editing.

Its own, not the neighbour's. `Sync-AudionBackup.ps1` used to take `rclone.exe` and the R2 keys from the publisher — "so as not to keep a second copy of the same keys". The price of that economy: without the publisher beside it the mirror did not come up at all, while it is the thing that has to come

up first on an empty machine. Now the program has its own binary in `Tools\rclone` and its own keys in `config\rclone`, and `Engine\BackupRclone.psm1` knows exactly three things about them: where to look, what to configure with, how to call.

The Build-stack snapshot moved out of the publisher. `backup.cmd` and `Backup-AudionArchives.ps1` live here now; in the publisher (version 1.8.0) the "БЭКАП" button is gone. The split follows the direction of the action: the publisher deals with what leaves for the outside, and everything that puts copies away and fetches them back is gathered in one place. The operation itself did not change by a line.

The window. Three thirds with draggable bars: mirror folders and state on the left, machines in the middle, the terminal down the right. The buttons do no work themselves — each calls the same script a person would call from a console, so the window and the wrappers never drift apart. English captions, Russian tooltips, a 1500 ms tooltip delay — as in the orchestrator, whose frame this borrows; the child-process machinery comes from the publisher.

State dots. The machine buttons carry four meanings and no "probably": green — checked and answering, amber — present but not raised, red — checked and silent, grey — nothing to check. Services are read on their own every thirty seconds; the network is probed by a button and once at open. The probes leave together and are awaited by the window's timer: six connections in a row would freeze the window for three seconds. The addresses come from the same `targets.json` the wrappers read — there is no second list of addresses in the program.

The header badges show the data folder and the release root. The second one turns amber when the chosen folder holds no private archives: `BACKUP` would refuse on such a folder, and seeing that before pressing is better than after. Long paths are trimmed from the head — what tells them apart is the tail.

Keys. `export-keys.cmd` collects this machine's keys into `Tunnel\Export`, `import-keys.cmd` lays them into the three places programs look for them: the profile, `config\rclone`, `core.sshCommand` of the mirrors. A key is a file with a matching `.pub`, so any naming travels. Private copies get their permissions narrowed at once: a key reachable by anyone but its owner is silently rejected by `ssh`, and that is the most common reason why "nothing works" on a new machine. Permissions are fixed in the source as well: the routes written by the tunnel installer point into the store.

Paths repair themselves. The import rewrites only those paths in `~\.ssh\config` that lead nowhere and leaves working lines alone. Moving the program proved it on the first run: `ProxyCommand` pointed at the old location of `cloudflared` and was corrected without a single manual edit.

Icon and launcher. `Start.exe` is built by the compiler that ships with Windows (`Engine\Launcher`), with `Build-Icon.ps1` beside it. Every icon frame is drawn in its own pixels rather than shrunk from a large one: at sixteen points the halo turns to mud. The glow is kept only for frames of 64 and above.