Audion Get Tools - user guide

Contents

Audion Get Tools installs, updates, and removes Windows software through WinGet: you tick what you need and press one button. This guide covers what is worth knowing while working with it - how to fetch an installer without installing anything, how to sort out the Visual C++ runtimes, and how to use AI-assisted package planning.

The project layout, the services, and the development rules live in the README. That is the technical documentation; it is not repeated here.

The WinGet catalog: sets, install, update

The main window is the catalog: 126 packages in twelve blocks - System, Developer tools, AI, PKMS and notes, Office and documents, Media (images, audio, video), Browsers, VPN and communication, Hardware and utilities, MSVC runtime 2015+ and Advanced: MSVC 2005-2013. A block is a grid of cards with checkboxes; above the list sits a filter by name and ID, every block has `Select block` and `Clear block`, and the `Selection profiles` row on top (Clear, Minimal, Dev, Media, Network) ticks a typical set in one press, after which any card can be changed. `Export YAML` and `Import YAML` in the navigation row save the current ticks to a file and bring them back, even on another machine.

Every block is a plain list of IDs in the configuration: your own program is added as a line in the file or with `Add ID to list` from the `Install / uninstall by ID` window, which also offers `Search` through the WinGet catalog and installing or removing a typed ID.

Window	What it does
`Install selected`	Scans the system first and shows only what is missing, then installs the ticked IDs without the Y/N prompts, with live progress in the log.
`Preview updates`	Shows the `winget upgrade` list as "name, ID, current, available" and changes nothing.
`Update selected`, `Update available`, `Update all available`	Updates by catalog block, by the WinGet list with checkboxes, or everything at once.
`Pin selected`	Keeps `config\pins.txt`: ticked installed packages get a blocking pin and stop updating until unpinned.
`Uninstall selected IDs`	What is installed, by the same blocks; system and runtime IDs are protected and take an extra checkbox.
`Update WinGet`	Updates the package manager itself; restart the program afterwards.
`Import / export`	`winget export` to JSON from a configured machine and `winget import` on a new one, with checkboxes about versions, upgrading installed packages and unavailable ones.

Everything that changes the system asks for confirmation before it runs; downloads and checks run at once.

Downloading without installing

Every program in the list has small buttons to the right of its name. They do not touch the checkbox: pressing a button neither selects nor clears the entry, so the usual workflow stays the same.

Button	What it does
Green down arrow	Downloads the program's installer into `output\Downloads` and installs nothing.

Button	What it does
Crimson box	Downloads the zip archive or the standalone build - the one that needs no installation. It appears only for programs that really have such a build.
Blue up arrow	Opens the download page in your browser: the GitHub releases page, the TechPowerUp page, or the vendor's own site.

The crimson button shows up on about a quarter of the list - 29 programs. Every package was checked for an archive build, and the button stays hidden where there is none, so it is never a dead press. Notepad++, Everything, OBS Studio, VLC, Telegram, FFmpeg, Node.js, MKVToolNix, Audacity, Sysinternals, Rufus, yt-dlp, SumatraPDF, Tabby, Rclone, and RClone Manager are among them.

Sometimes the vendor ships a build that never reaches the WinGet catalogue - Tabby and RClone Manager are two. The file is then taken straight from the release page, and the button behaves exactly as it always does. Both buttons draw from the same release, so the installer and the portable build are always the same version. For RClone Manager it is also the newer one: the WinGet catalogue carries the installer alone, and it trails what the vendor has already published.

The same helps when WinGet simply cannot hand out a file: if the program has a GitHub release page, Audion Get Tools looks for the build there. Every check stays in place - other systems, the wrong architecture, checksums and debug files are refused - and when no single file clearly fits, the button says so instead of downloading something at random.

Why this is useful when a normal install is right there:

- an installer can be carried to another machine or kept for work without a network;
- some programs do more when installed from their own distributive than from their WinGet package. PowerShell from the Microsoft installer adds Explorer context-menu entries and keeps updating itself afterwards, while the WinGet build does not;
- on the download page you can pick a build by hand: a different architecture, a portable edition, an earlier release.

`Google Chrome (web installer)` has its own download button next to it - it fetches the same web installer without running it.

Where things land:

Folder	What is there
`output\Downloads`	Installers, loose - no subfolders. You pick one up, run it, and forget it.

Folder	What is there
`output\Portable\<Program>`	Whatever runs without installation: archives and standalone builds, each in a folder named after the program - `output\Portable\Notepad++`, `output\Portable\Rufus`.
`output\Install\<Bundle>`	Whatever installs into the system, for example `output\Install\MSVC All-in-One (TechPowerUp)`.

The folder is named the way the vendor writes the product name. No `Google.Chrome`, no underscores. Inside that folder everything keeps the names it arrived with.

The split follows what a download does, not its extension: the MSVC bundle installs runtimes into the system, so it lives in `Install` even though it arrives as a zip - unpacked next to its archive so `install_all.bat` is at hand.

All of it sits under `output` and is wiped with it when the workspace is cleaned - by design: a portable build should not carry gigabytes of installers around. Move what you need somewhere safe; the rest can simply be downloaded again.

Vendor Downloads

WinGet hands out the current release of a program. The `Vendor Downloads` section reads the vendor's own catalog instead and lists every version it ever published - with the release date, newest first. That is how a version store is kept: the fresh one and the one everything worked on.

It is one screen for every vendor: the tabs at the top pick whose catalog to browse, the active one underlined in blue; Windows has a second tab row under them, `Version`, `Apps`, `Drivers`, because it has more fields than one screen holds, while the options row above is shared; on the `Version` page the product, platform, edition and language sit right above the cards, and a change of any of them shows in the build captions at once; right under them the options sit in one row (unpack the zip, delete the archive, build the ISO, flat folder), below come the platforms and products, and the version list and the buttons are shared. Every block opens with a row of macro buttons: `Image kind` (Business, Consumer, Pro only) and `App set` for Windows, `Pick` above the version cards for everyone.

Vendor	What is there
Windows (UUP dump)	Every Windows 11 and Windows 10 build in the UUP dump catalog, each cumulative update as its own build number with its date, plus Insider; x64 and ARM; language and edition as toggles. A script package is downloaded and the ISO is built on this machine from Microsoft's own files.

Vendor	What is there
TechPowerUp	The site's catalog by section, every entry with its whole version history, date and size. Drivers: AMD Radeon and Ryzen chipset, Intel graphics (Arc, iGPU), Wi-Fi, Bluetooth, Ethernet and NPU, Qualcomm Snapdragon X. Utilities: DDU, NVCleanstall, NVIDIA Profile Inspector, ThrottleStop, DRAM Calculator for Ryzen, Samsung Magician, Visual C++ Runtimes AIO, MemTest64. Monitoring: GPU-Z, CPU-Z, AIDA64 Extreme, ZenTimings, Real Temp. Benchmarks: 3DMark, PCMark 10, Cinebench, FurMark, Prime95, Unigine Superposition and Heaven, GravityMark, Linpack Xtreme, ATTO Disk Benchmark. Video BIOS: NVIDIA NVFlash, AMDVBFlash. The file comes through TechPowerUp's signed link. A version is one card: the green arrow fetches the installer, the crimson archive icon the portable build of the same version, an empty slot means there is no such file; the Portable button keeps the versions with a portable build and a tick then means that file. NVIDIA drivers and DLSS stay on the NVIDIA tab.
NVIDIA	Game Ready and Studio drivers by chip generation: RTX 50, 40, 30, 20, GTX 16, 10, 900, 700, 600, 500, 400, desktop and notebook. NVIDIA App and Broadcast - the current installers. CUDA Toolkit - every release from the archive. DLSS, DLSS Frame Generation and DLSS Ray Reconstruction - every library version from TechPowerUp.
Blackmagic Design	DaVinci Resolve and Resolve Studio, Fusion, Blackmagic RAW, Desktop Video, Camera Setup and the rest of the catalog - 45 products for Windows, Windows ARM, macOS and Linux.
Affinity	Affinity 3 by Canva - the unified free app, current release only, exe and msix for x64 and ARM, dmg for macOS. Photo 2, Designer 2 and Publisher 2 - every 2.x release from the Serif updates pages, each Windows version as msix and as exe.

Windows through UUP dump is the way to get an image "as of a given month", updates already inside, without waiting for Microsoft to refresh its media. Microsoft ships the system as the same UUP files Windows Update installs, and UUP dump keeps a catalog of everything it has seen. A version card here is a build, for example `24H2 26100.9278 – 27 Aug 2026`; type a release or a month into the filter. What is downloaded is a small package with `uup_download_windows.cmd`, and there are two ways from there:

- the `Build the ISO right away` checkbox: after unpacking, the script starts in its own console window, downloads the files from Microsoft's servers with aria2 and assembles a bootable ISO. The build does not run under the destination but in a `UUP` folder at the root of the drive the program sits on (`E:\UUP`, say): the script needs a short path without spaces, and the program creates it by itself. The finished image is moved to the destination folder, into `Vendors\Windows (UUP dump)` or straight into it with `Straight into the destination folder` on; when an image of that name is already there, the new one goes into a subfolder named by date and time, the old one is never overwritten. The script needs administrator rights, which it asks for itself;

- the `Delete cache after ISO` checkbox appears together with the previous one: once the image is moved, the build folder in `UUP` with the scripts and the 9 GB of Microsoft's files is deleted. Without it the folder stays, and the same build can be reassembled with another app set without downloading again;
- the `Embed drivers from input into the image` checkbox in the options row at the top of the tab. The order is: the `This machine's drivers` section on the same tab shows every third-party driver package as a card, the network ones (Wi-Fi, Ethernet, Bluetooth) ticked already, enough for a fresh system to get online while Windows Update brings the rest; tick more if you need them and press `Export drivers` in the navigation row, each in its own folder with .inf, .sys and .cat; move the `Drivers` folder anywhere inside `input` (the Source path at the top of the window), under any name and at any depth, the program finds every `.inf` by itself; tick the box and build. The packages are copied into the build's `Drivers\OS`, the converter switches `AddDrivers` on and DISM embeds them into install.wim, so Setup installs them with the system. Only unpacked INF packages work, a vendor's .exe installer does not; the architecture must match the image; unsigned drivers need test signing. Storage and RAID drivers that Setup itself needs to see the disk are not covered;
- without the checkbox the package simply sits in the build folder, and the script can be run whenever, even on another machine.

The package is requested with updates integrated, .NET 3.5 included and cleanup on.

Updates and extra editions are put into the image by DISM, and the converter takes it from the Windows ADK Deployment Tools. Without the ADK it falls back to the system DISM, which on Windows 11 25H2 cannot service the image: the build then quietly comes out without updates and without editions, a bare base. So while the ADK is not installed the download buttons in this section are dimmed with the reason in the tooltip, and there is an `Install ADK` button below: it fetches adksetup.exe from Microsoft and installs only the Deployment Tools, about 100 MB, asking for administrator rights. Once installed, the buttons open by themselves. And if the converter still fails at something, the program reads its error log and says so instead of calling the build done.

Above the edition sit three `Image kind` presets: `Business` sets Pro with Enterprise, Education, Pro Education and Pro for Workstations, as in the business editions ISO; `Consumer` sets Home and Pro in one image with Education, Pro Education and Pro for Workstations, as in the consumer editions ISO; `Pro only` clears the extras. The `Edge (1)` button in the apps block decides whether the Edge browser goes into the image. The Edge WebView2 runtime is untouched and stays in the system: it is a separate component, apps run and are built on it, from the new Outlook and Teams to third-party programs.

The `Built-in apps` block decides which Store apps go into the image. `As Microsoft` puts in the stock set, `None` puts in none, `Chosen below` (the default) opens a list of 59 cards where ticked apps are installed and the rest never appear in the system, so there is nothing to remove afterwards. Three `App set` presets fill the list: `Minimal` (the four system packages), `Work` (plus thirteen tools and ten codecs) and `Everything`; each set is laid out in the table below. The `Include in the distribution build` row gathers the ballast into toggle buttons: a button that is not pressed keeps its group out of the image, a pressed one installs it. By default all seven are off, and the image ships with the `Work` set and without Edge. The number in the label is how many packages the group holds, the list sits in the button's tooltip: `Media stack (4)` is Media Player, Films & TV, Photos and Clipchamp; `Edge (1)` is the browser itself, there is no separate checkbox for it; then `Xbox (6)`, `Teams and mail (6)`, `Bing and widgets (6)`, `Promo and helpers (8)` and `Small tools (2)`. The buttons and the cards below show one and the same choice: untick one card of a group and its button goes dark. The mechanism is the same for every Windows 11 client edition, not only N: the converter comments the surplus lines out of its package list and leaves them out of the image. OneDrive and Copilot are not Store packages; they are removed after installation.

Install sets, the buttons of the `App set` row; each one just fills the card list, which can then be edited by hand or with the group buttons:

Set	Packages	What gets installed	Who it is for
`Minimal`	4	system only: Store, Store purchases, Windows Security, App Installer	an image for an application server or a kiosk that needs nothing from the Store while WinGet and Store updates keep working
`Work`	27	the system packages, thirteen tools (Notepad, Windows Terminal, Calculator, Snipping Tool, Paint, Camera, Sound Recorder, Phone Link, Clock, Sticky Notes, To Do, Weather, Power Automate) and ten codecs	a work machine without entertainment and promotion; the default, and the reference image was built with it
`Everything`	59	the whole catalog as Microsoft ships it, only with the list open for editing	when everything stock is wanted with the option to untick a card or two

The `Image kind` row above is about editions, not apps: `Business` is Pro plus Enterprise, Education, Pro Education and Pro for Workstations, `Consumer` is Home and Pro plus Education, Pro Education and Pro for Workstations, `Pro only` is one edition. Sets and image kinds combine freely.

The whole Store package catalog the way the program divides it: the `Work` set plus the seven buttons of the `Include in the distribution build` row. 59 packages in total plus Edge, nothing sits outside a group.

Group	Count	Inside	By default
System	4	Microsoft Store, Store purchases, Windows Security, App Installer (WinGet)	in the image, `Work` set
Tools	13	Notepad, Windows Terminal, Calculator, Snipping Tool, Paint, Camera, Sound Recorder, Phone Link, Clock, Sticky Notes, To Do, Weather, Power Automate	in the image, `Work` set
Codecs	10	Web Media, RAW images, HEIF, HEVC, VP9, WebP, AV1, MPEG-2, AVC encoder, Dolby Audio	in the image, `Work` set
Media stack	4	Media Player, Films & TV, Photos, Clipchamp	off
Edge	1	the Edge browser only; the Edge WebView2 runtime is untouched and stays, apps run and are built on it	off
Xbox	6	Xbox app, Xbox Game Bar, Xbox Game overlay, Xbox speech overlay, Xbox identity, Xbox TCUI	off
Teams and mail	6	Teams, Outlook (new), Mail and Calendar, People, Office hub (M365), Family	off
Bing and widgets	6	Bing Search, News, Widgets (web experience), Widgets runtime, Start experiences, Cortana	off
Promo and helpers	8	Solitaire, Tips, Get Help, Feedback Hub, Quick Assist, Dev Home, PC Manager, App compatibility enhancements	off
Small tools	2	Maps, Cross Device	off

The `Minimal` set is the four system packages without the tools and the codecs, `Everything` turns every group on at once. OneDrive and Copilot are not in the catalog: they are not Store packages and are removed after installation.

On editions, plainly: through UUP Microsoft ships only the consumer Home and Pro with their N variants, plus Team for Windows 10. Everything else UUP dump derives from Pro while building, and that is the `Additional editions in the ISO` block: Enterprise, IoT Enterprise, IoT Enterprise K, Education, Pro Education, Pro for Workstations and Enterprise multi-session. Ticked ones are added to

the same image next to the base edition, and the version card shows them in its caption. LTSC cannot be had this way - it is neither in the catalog nor among the derived editions; Microsoft ships LTSC through corporate channels only. ARM is in the catalog for the same builds, the platform toggle.

NVIDIA drivers are picked by chip generation, not by the exact card: one driver file covers several generations, and the list shows what NVIDIA publishes for the chosen one. `My cards` keeps only the versions published for every generation ticked in the `My generations` row that appears under the switch - one driver for every card in the house; the ticks are remembered in `config\vendor_nvidia.yaml` and come back on the next start. Every version carries its date, size and marks:

- `NVENC SDK 13.1` - the nv-codec-headers generation an FFmpeg build was made against: such a build works on this driver and on any newer one; SDK 13.1 wants 610, SDK 13.0 wants 570, SDK 12.2 wants 551.76;
- `FFmpeg <= 8.0.1` - the newest FFmpeg build from Audion Media Tools whose hardware encoding works on this driver: 9.0.1 wants 610, 8.0.1 and 7.1.1 want 570, 7.1 wants 551.76;
- `CUDA <= 13.0` - the newest CUDA Toolkit the driver supports. CUDA is backwards compatible: the driver also runs every older toolkit, and a program built on CUDA 12.6 works with any driver from 560.76 on;
- `★ golden` - versions the community treats as keepers; that card is gold.

All lists live in `config\vendor_nvidia.yaml` and are edited by hand: add your own keeper, a threshold for a new SDK or the generations of your cards, and the list shows it the next time it opens.

DLSS is not downloaded as a program: it is the `nvngx_dlss.dll` library every game carries, and the driver and the NVIDIA App can swap it for a newer one. Individual versions of the three libraries - upscaler, frame generation, ray reconstruction - are here to drop into a game folder or keep at hand for the swap.

How it works:

- platform and product are toggles; `Whole catalog` for Blackmagic and `All four` for Affinity show every product at once with the product name in front of each version, and the filter box narrows the list;
- versions are cards with checkboxes. Tick the ones you need and press `Download` below - or press the button on a card to take that one build. The `Pick` row above the cards ticks them by rule: `Latest stable` takes the newest final release, skipping betas and Insider builds, `★ Golden` every starred card, `Clear` empties the choice. The star and the gold frame go to NVIDIA drivers from the golden list and, for Windows, to the newest build of each generation, 22H2, 24H2, 25H2

and so on, with its latest cumulative update; the other builds of a generation are history, and in a list of hundreds the gold ones stand out at once. The link button only prints the link to the log and to `report\links.txt`, downloading nothing;

- every build lands in its own folder, `output\Vendors\<Vendor>\<archive name>`, for example `output\Vendors\Blackmagic Design\DaVinci_Resolve_Studio_21.0.4_Windows` or `output\Vendors\Affinity\affinity-photo-2.6.5`. When the destination folder already is your store for that vendor, enable `Straight into the destination folder`;
- an interrupted download resumes: run the same build again and the file continues where it stopped. A finished file is never fetched twice;
- a zip is unpacked into its folder so the installer is at hand, and is deleted after unpacking by default. Both are checkboxes. exe, msix, dmg and tar stay as they are.

No account is needed anywhere but one place: the free DaVinci Resolve is issued by Blackmagic only after a form (name, e-mail, phone, address). The form fields appear at the bottom of the section while that product is selected; they go straight to blackmagicdesign.com, no account is created, no confirmation mail follows, and nothing is stored here. Signed Blackmagic and Serif links live a few hours; the Affinity 3 links are permanent.

This machine's drivers

The `This machine's drivers` section sits on the Windows (UUP dump) tab of Vendor Downloads, between the build and the versions. `pnputil /enum-drivers` builds the list: one card per third-party package, the class first (Network, Bluetooth, Chipset / system, Storage / RAID, Audio, Display and so on), then the original inf name, the provider and the version; Microsoft's in-box drivers are not shown. The Network and Bluetooth classes are ticked when the window opens, and only the newest version of each driver, older copies from the store stay unticked: enough for a fresh system to get online, and Windows Update then brings the remaining drivers, newer ones at that. Want more, tick the chipset, storage, audio, even the GPU, minding that install.wim grows by every package. The `Export drivers` button in the navigation row runs `pnputil /export-driver` for every tick and lays the packages down as folders with .inf, .sys and .cat; nothing on the machine changes. That folder then goes either into `input` to be embedded into an image, or onto a fresh system with `pnputil /add-driver <folder>\*.inf /subdirs /install`.

Service buttons for outside tools

The Windows (UUP dump) section carries `WinUtil` in the navigation row next to the ADK button: Chris Titus's Windows utility, tweaks, debloat, program installs, Windows updates control. It opens in a

window of its own: Windows Terminal when installed, otherwise a plain console, with PowerShell 7 when present and Windows PowerShell 5.1 as the fallback. The script is downloaded from the author's site on every launch and asks for administrator rights itself; the program does not wait for it. The service procedures keep `Windows licence state`: read-only, edition, licence channel and expiry through the stock slmgr, printed to the log.

Visual C++ (MSVC) bundles

Old games and programs need Visual C++ runtimes from various years. This is usually the messiest part of a Windows system, so it is laid out in three places.

Where	What is there
`MSVC runtime (2015+)`	Official Microsoft packages, x86 and x64. A modern system needs nothing else.
`Advanced: MSVC 2005–2013`	Official packages, one per year and architecture - for ancient software only. They never report an update: the vendor froze those builds.
`Classic scripts`	Two ready-made bundles: `MSVC All-in-One (TechPowerUp)` and `MSVC AIO (abbodi1406)`. Each has a download button and a button that opens its release page.

The bundles beat installing one package at a time: a single pass covers every runtime and includes 2012, which has no WinGet package at all. They are assembled and kept current by well-known maintainers, so for the older family they are the better choice.

The `Check MSVC runtimes` action shows what is actually installed, which version is available, and whether anything needs doing. Start there when it is not clear what is missing.

Installing runtimes changes the system, so Audion Get Tools asks for confirmation first. The application has to run as administrator - without that the install does not start and you get a message saying so.

Removal is not split into groups: all nine runtimes go at once. That is deliberate - this set is safer to remove completely and reinstall clean than to repair piece by piece.

AI Package Planner

The Audion Get Tools section that uses an LLM to suggest Windows/WinGet packages, validate exact IDs, and execute selected actions through the normal protected Audion paths.

Purpose

`AI Package Planner` lets a user describe a software task in natural language, get a short package plan, validate suggestions with WinGet search, and then run only selected exact IDs.

The section is not a separate installer. It must not bypass the existing install, update, pin, check, or uninstall flows. The LLM helps draft and explain a reviewable plan.

Primary value:

- suggest packages from a human request;
- reduce WinGet package ID mistakes;
- show a plan before installation;
- support focused search and exact-ID actions;
- keep keys, models, and prompt templates inside the GUI-managed flow.

Main Workflow Model

The section has two tabs:

Tab	Purpose
`Planner`	AI task, instruction template, instruction editor, plan building, WinGet search, and selected exact-ID execution.
`Models`	OpenAI/Gemini provider selection, keys, models, manual overrides, reasoning, and model checks.

Top-right CTA in the section navigation:

`RUN SELECTED WITH AI`

It opens the final form for running selected packages from the last AI plan. This is the main user CTA after a plan has already been generated and reviewed.

`Build plan` stays as the first button in the `Actions` grid because it starts planning; it is not the final execution step.

Planner Fields

AI task

What to find or prepare right now.

Examples:

- `Find Media Player Classic Black Edition with WinGet.`
- `Prepare a Windows machine for Dev/Media.`
- `Suggest tools for screen recording, editing, and video conversion.`

This field is required for building an AI plan. Without it, the LLM does not know what to suggest.

Instruction template

A saved AI behavior template.

Selecting a template loads text into `AI planner instruction`. It does not scan installed packages, call the LLM, or install anything.

Buttons beside the select:

Icon	Action
`sync`	Refresh template list.
`save`	Save the current instruction to prompt cache.
`push_pin`	Pin the selected template.
`block`	Unpin the selected template.
`delete`	Delete the selected template from cache.
`backspace`	Clear the instruction editor.

AI planner instruction

Rules for how the AI should reason and format the plan.

Most users do not need to edit this field every time. They can select an instruction template or keep the default prompt.

The instruction defines behavior:

- propose options first;
- do not install anything during planning;
- validate or clarify WinGet IDs;
- mention risks and ambiguous choices;

- keep the answer short and practical.

Scan installed IDs

Sends already installed WinGet IDs to the LLM.

This helps avoid suggesting packages that are already present. Installed state is based only on the installed-ID scan, not on known Audion groups.

Send Audion known package groups

Sends curated project lists to the LLM as context.

Important: known Audion groups are options and presets, not installed-state proof.

Search candidates

How many `winget search` rows to collect for each suggested package.

Smaller values are faster and cleaner. Larger values are useful for ambiguous searches.

Planner Actions

Action	Behavior	Type
`Build plan`	Sends the task and instruction to the LLM, then validates candidates with WinGet search.	Safe
`Search WinGet`	Searches by name, task, or exact ID and writes a report.	Safe
`Run selected from last AI plan`	Opens the form for selected exact IDs from the last plan.	Dangerous for install/update/uninstall
`Add exact ID to list`	Adds a reviewed ID to a config list and GUI group.	Safe
`Install exact ID`	Installs one exact ID through the normal Audion path.	Dangerous
`Update exact ID`	Updates one exact ID through the normal Audion path.	Dangerous
`Uninstall exact ID`	Uninstalls one exact ID through the protected Audion path.	Dangerous

Dangerous actions use the standard Audion confirmation flow.

Building An AI Plan

When `Build plan` is clicked, the service:

1. Reads `AI task`.
2. Reads the prompt from `AI planner instruction` or the selected template.
3. Optionally scans installed WinGet IDs.
4. Optionally adds known Audion groups as context.
5. Calls the selected LLM provider.
6. Requires a JSON response following the package-plan schema.
7. Normalizes model suggestions.
8. Validates candidates with `winget search`.
9. Saves the last plan to cache.
10. Writes Markdown/JSON reports to `report\`.

The LLM returns a structured object with summary, packages, and notes.

Each package can include:

- display name;
- search query;
- proposed or exact WinGet ID;
- Audion group;
- intended action;
- reason;
- risk or review note.

WinGet ID Validation

The planner does not trust IDs only because the LLM suggested them.

Validation statuses:

Status	Meaning
`validated_exact`	Exact ID found through exact WinGet search.

Status	Meaning
`single_search_candidate`	No ID was provided, but search produced one strong candidate.
`search_candidates`	Candidates were found and need human review.
`id_not_found`	Suggested ID was not found.
`review_required`	Manual review is required.

Only usable exact IDs are exposed as checkboxes for running the AI plan.

Running Selected IDs From The AI Plan

The final flow opens from the top CTA `RUN SELECTED WITH AI` or the `Run selected from last AI plan` button in the action grid.

The form contains:

- `Action`;
- `Last AI plan exact IDs`;
- checkbox YAML import/export;
- final run button for the selected operation.

Available actions:

Action	Behavior
`Install missing`	Installs selected exact IDs through batch install.
`Update installed`	Updates selected exact IDs through batch update.
`Pin installed`	Adds IDs to pins and applies blocking pins.
`Check installed`	Checks selected IDs.
`Uninstall selected`	Uninstalls selected exact IDs through batch uninstall.

This path uses `winget_service._run_package_batch`, so it stays inside Audion's shared package-action logic.

Exact-ID Actions

The section also supports actions that do not require an LLM plan.

Search WinGet

`Search query` accepts:

- app name;
- vendor name;
- task;
- exact WinGet ID.

`Exact ID search` switches the search to exact mode.

Results are written to `report\<timestamp>_ai_search_winget\winget_search.md` and JSON.

Add exact ID to list

Adds a reviewed package ID to a selected list:

- `system`
- `dev`
- `ai`
- `pkms`
- `office`
- `media_images`
- `media_audio`
- `media_video`
- `network`
- `hardware`
- `custom`
- `pins`

For GUI-backed groups, the ID is added to `config\tool_manifest.yaml` so it appears as a normal checkbox.

Install exact ID

Runs the normal Audion install-by-ID path.

Update exact ID

Runs batch update for one ID.

Uninstall exact ID

Runs the normal protected uninstall-by-ID path.

Models

The `Models` tab manages provider and model selection. It does not install packages.

Fields:

Field	Purpose
`Model family`	Switches OpenAI/Gemini.
`OpenAI API key` / `Gemini API key`	Selects a key reference from config or cache.
`OpenAI model` / `Gemini model`	Selects a cached or API-discovered model.
`Model override`	Manual model ID if the model is valid but not in the list yet.
`OpenAI reasoning`	Reasoning effort for supported OpenAI models.
`LLM max output`	Response size limit.
`LLM retries`	Retry count after failures.
`LLM timeout`	Maximum wait for one request.

`Check model` sends a small request and stores the status in cache.

Caches And Local Files

File	Contents
`config\api_key_openai.txt`	Local OpenAI API key.
`config\api_key_gemini.txt`	Local Gemini API key.
`config\llm_settings.yaml`	Provider settings and defaults.
`config\gui_key_cache.json`	Favorite key references, no key material.
`config\gui_model_cache.json`	Model lists, favorites, and check statuses.
`config\gui_package_prompt_cache.json`	Saved prompt templates and pins.

File	Contents
`config\gui_package_plan_cache.json`	Last AI package plan.

API key material must not be written to GUI caches, logs, reports, or release archives.

Reports

AI Package Planner writes reports under `report\`:

- `ai_package_plan.md`
- `ai_package_plan.json`
- `winget_search.md`
- `winget_search.json`
- `action_summary.md`
- `action_summary.json`

Operation logs are written under `logs\`.

Safety Rules

Core rules:

- The LLM does not execute installation itself.
- The plan is shown to the user first.
- Only selected exact IDs are executed.
- Dangerous actions use Audion confirmation.
- WinGet IDs are validated through search before plan execution.
- Known Audion groups are not treated as installed state.
- API keys are not written to caches or reports.
- Protected uninstall remains inside the normal Audion protected flow.

Typical User Flow

1. Open `AI Package Planner`.
2. On `Planner`, fill `AI task`.
3. Keep or choose `Instruction template`.
4. Edit `AI planner instruction` only if needed.

5. Click `Build plan`.
6. Review the log and plan report.
7. Click `RUN SELECTED WITH AI`.
8. Choose an action and exact IDs from the last plan.
9. Confirm execution.

Typical DevOps Flow

1. Configure `config\api_key_openai.txt` or `config\api_key_gemini.txt`.
2. Check the model on `Models`.
3. Pin the working model and key.
4. Create and pin a prompt template for the workflow.
5. Build plans with `Build plan`.
6. Add reviewed IDs to project lists with `Add exact ID to list`.
7. Execute install/update only through Audion exact-ID operations.

Troubleshooting

Symptom	Check
Model does not answer	Key, selected model, timeout, max retries.
No models in list	API key, live model request, cache refresh.
Empty plan	`AI task`, prompt, LLM availability.
ID not found	`Search WinGet`, exact ID, source.
Package not offered for execution	Validation status, exact ID, last plan cache.
Prompt does not change	Selected `Instruction template`, cache refresh, clear button.

Developer Entry Points

File	Purpose
`config\tool_manifest.yaml`	Command, field, tooltip, and GUI group definitions.
`system_core\ui_nicegui\app.py`	Planner/Models tabs, action grid, top CTA, and pending forms.

File	Purpose
`system_core\services\winget_ai_service.py`	LLM calls, prompt/model/key cache, WinGet search validation, plan reports, exact-ID actions.
`system_core\providers\openai_provider.py`	OpenAI JSON/structured calls.
`system_core\providers\gemini_provider.py`	Gemini structured calls.
`system_core\services\winget_service.py`	Shared Audion install/update/pin/check/uninstall paths.

MCP/WinGet Boundary

The section is designed as an AI layer over WinGet: plan first, validate, let the user choose, then execute exact IDs.

Even as WinGet MCP support evolves, the safety model remains:

- plan first;
- validate exact IDs;
- user review;
- execute after approval;
- never bypass Audion package paths.

What WinGet itself ships today (checked against 1.29.280)

`winget mcp` prints the JSON snippet that wires `WindowsPackageManagerMCPServer.exe` into any MCP client. The server (`winget-mcp`) exposes exactly two tools:

- `find-winget-packages` — search and upgradeable listing (`query`, `upgradeable`), read-only;
- `install-winget-package` — install or upgrade (`identifier`, `source`, `upgradeOnly`), destructive.

Both are a subset of what Audion Get Tools already does directly: search for plan validation, and install/update by exact ID. Routing through MCP would drop the live ConPTY progress, the logs and reports, and the protected-ID check, so the planner keeps calling WinGet itself. `Health / Doctor` reports the MCP server path so it can be attached to an external MCP client without changing how the app runs packages.

Review Before Execution

For every selected package, verify exact ID, publisher, source, requested action, and whether the command affects the current user or the whole machine. Remove ambiguous candidates and packages already managed by another installer or corporate policy.

Use scan results to distinguish installed, available, upgradable, missing, and unknown IDs. Do not infer successful installation from a search result or planner selection.

After A Run

Review exit codes and the report for each ID. Some installers return success but require a reboot or interactive completion. Confirm the executable or registered package version when the item is important to an Audion runtime.

Keep failed IDs in a separate retry list with the original source and error. Rerun only after checking network, source availability, elevation, architecture, and installer conflicts.

GUI Manifest And Documentation

The GUI manifest defines the visible planners, package actions, fields, defaults, help text, and command bindings. This guide translates them into safe package-management decisions. When the manifest adds or renames an action, update the relevant workflow here and preserve exact-ID, source, elevation, review, and rollback cautions.

For a release check, compare the manifest labels with the GUI, command preview, generated package list, and final report. A friendly label must never conceal whether the backend searches, downloads, installs, upgrades, removes, or only stages a command.

Reference: every window, every control, every tooltip

This section is generated from `config\tool_manifest.yaml` by `tools\docs\build_control_reference.py`, so the captions and tooltips here are the window's own, letter for letter. Windows follow the order of the Operations list. Every control shows its type, default, tooltip (what pops up on hover) and, where present, its options and the condition that shows it.

Install selected

Scan the system, show only missing packages, and install checked IDs without Y/N/Q prompts.

Fields

Selection profiles — a row of preset buttons: a press fills the fields below.

- Note under the field: Presets only mark checkboxes; you can adjust every package before running.
- Buttons:
  - **Clear**
  - **Minimal**
  - **Dev**
  - **Media**
  - **Network**

System — cards with checkboxes, any set.

- Default: empty
- Options (13 items): .NET Framework 3.5 (Windows feature); .NET Desktop Runtime 6; .NET Desktop Runtime 8; .NET Desktop Runtime 10; Windows Terminal; PowerShell 7; Tabby; 7-Zip; PeaZip; WinRAR; Process Explorer; Autoruns; TeraCopy

Developer tools — cards with checkboxes, any set.

- Default: empty
- Options (25 items): Everything; WinMerge; Notepad++; yt-dlp; RHash; Git; Visual Studio Code; VSCodium; GitHub Desktop; GitKraken; FFmpeg; btop4win; PowerToys; Total Commander; Far Manager; grepWin; Python 3.12; Python 3.13; Node.js; JetBrains Mono Nerd Font; JetBrains Toolbox; IntelliJ IDEA Community; PyCharm Community; WebStorm; ShareX

AI — cards with checkboxes, any set.

- Default: empty
- Options: Claude; Claude Code; OpenAI Codex

PKMS and notes — cards with checkboxes, any set.

- Default: empty
- Options: Notion; Notion Calendar; Obsidian; Joplin; Evernote; UpNote; Zettlr; AppFlowy

Office and documents — cards with checkboxes, any set.

- Default: empty
- Options: Calibre; LibreOffice / soffice; Acrobat Reader; SumatraPDF

Media: images — cards with checkboxes, any set.

- Default: empty
- Options: FastStone Viewer; Krita; GIMP; Inkscape; ImageMagick; IrfanView; XnView MP

Media: audio — cards with checkboxes, any set.

- Default: empty
- Options: foobar2000; REAPER; Audacity; Ocenaudio

Media: video — cards with checkboxes, any set.

- Default: empty
- Options (14 items): VLC; Shutter Encoder; LosslessCut; HandBrake; MPC-HC; MPC-BE; mpv.net; MKVToolNix; Subtitle Edit; OBS Studio; Bandicam; Bandicut; MediaInfo; qBittorrent

Browsers, VPN, messaging — cards with checkboxes, any set.

- Default: empty
- Options (24 items): Firefox; Opera; Vivaldi; Brave; Zen Browser; Ungoogled Chromium; Cent Browser; AmneziaVPN; v2rayN; Karing; Happ; Signal; Telegram Desktop; Discord; Bitwarden; KeePassXC; Thunderbird; Zoom; AnyDesk; Yandex Disk; Rclone; RcloneView; Rclone UI; RClone Manager

Hardware and service tools — cards with checkboxes, any set.

- Default: empty
- Options (15 items): CrystalDiskInfo; CrystalDiskMark; CPU-Z; GPU-Z; HWiNFO; Cinebench R23; FurMark 2; MSI Afterburner; Display Driver Uninstaller (DDU); NVCleanstall; HWMonitor; fastfetch; bottom; WizTree; Rufus

MSVC runtime (2015+) — cards with checkboxes, any set.

- Tooltip: Official Microsoft packages for the 2015+ (v14) runtime, x86 and x64.
- Note under the field: For a modern system this is the only runtime that matters. For the whole 2005-2013 family use the TechPowerUp or abbodi1406 all-in-one bundle in Classic scripts: it also covers 2012, which WinGet does not have.
- Default: empty
- Options: MSVC 2015+ x86; MSVC 2015+ x64

Advanced: MSVC 2005-2013 — cards with checkboxes, any set.

- Tooltip: Legacy runtimes for old software. WinGet manifests here are frozen at the final vendor build, so they never report an update.
- Note under the field: Official Microsoft packages, one per year and architecture. Recommended instead: the all-in-one bundles from TechPowerUp or abbodi1406 in Classic scripts - one pass, and they include 2012, which has no WinGet package.
- Default: empty
- Options: MSVC 2013 x86; MSVC 2013 x64; MSVC 2010 x86; MSVC 2010 x64; MSVC 2008 x86; MSVC 2008 x64; MSVC 2005 x86

Preview updates

Show the current WinGet update list as Name | ID | current -> available and write a report.

Update all available

Run winget upgrade scan and update every package with an available update.

Update WinGet

Update WinGet itself. Restart Audion Get Tools afterwards.

Tooltip: WinGet ships inside the Microsoft App Installer package and replaces itself while it runs, so it is updated alone, outside the package checkboxes. Restart Audion Get Tools after it finishes.

Update selected

Show available updates by project groups and update only checked IDs.

Fields

Selection profiles — a row of preset buttons: a press fills the fields below.

- Note under the field: Presets only mark checkboxes; you can adjust every package before running.
- Buttons:
  - **Clear**
  - **Minimal**
  - **Dev**
  - **Media**
  - **Network**

System — cards with checkboxes, any set.

- Default: empty
- Options (13 items): .NET Framework 3.5 (Windows feature); .NET Desktop Runtime 6; .NET Desktop Runtime 8; .NET Desktop Runtime 10; Windows Terminal; PowerShell 7; Tabby; 7-Zip; PeaZip; WinRAR; Process Explorer; Autoruns; TeraCopy

Developer tools — cards with checkboxes, any set.

- Default: empty
- Options (25 items): Everything; WinMerge; Notepad++; yt-dlp; RHash; Git; Visual Studio Code; VSCodium; GitHub Desktop; GitKraken; FFmpeg; btop4win; PowerToys; Total Commander; Far Manager; grepWin; Python 3.12; Python 3.13; Node.js; JetBrains Mono Nerd Font; JetBrains Toolbox; IntelliJ IDEA Community; PyCharm Community; WebStorm; ShareX

AI — cards with checkboxes, any set.

- Default: empty
- Options: Claude; Claude Code; OpenAI Codex

PKMS and notes — cards with checkboxes, any set.

- Default: empty
- Options: Notion; Notion Calendar; Obsidian; Joplin; Evernote; UpNote; Zettlr; AppFlowy

Office and documents — cards with checkboxes, any set.

- Default: empty
- Options: Calibre; LibreOffice / soffice; Acrobat Reader; SumatraPDF

Media: images — cards with checkboxes, any set.

- Default: empty
- Options: FastStone Viewer; Krita; GIMP; Inkscape; ImageMagick; IrfanView; XnView MP

Media: audio — cards with checkboxes, any set.

- Default: empty
- Options: foobar2000; REAPER; Audacity; Ocenaudio

Media: video — cards with checkboxes, any set.

- Default: empty

- Options (14 items): VLC; Shutter Encoder; LosslessCut; HandBrake; MPC-HC; MPC-BE; mpv.net; MKVToolNix; Subtitle Edit; OBS Studio; Bandicam; Bandicut; MediaInfo; qBittorrent

Browsers, VPN, messaging — cards with checkboxes, any set.

- Default: empty
- Options (24 items): Firefox; Opera; Vivaldi; Brave; Zen Browser; Ungoogled Chromium; Cent Browser; AmneziaVPN; v2rayN; Karing; Happ; Signal; Telegram Desktop; Discord; Bitwarden; KeePassXC; Thunderbird; Zoom; AnyDesk; Yandex Disk; Rclone; RcloneView; Rclone UI; RClone Manager

Hardware and service tools — cards with checkboxes, any set.

- Default: empty
- Options (15 items): CrystalDiskInfo; CrystalDiskMark; CPU-Z; GPU-Z; HWiNFO; Cinebench R23; FurMark 2; MSI Afterburner; Display Driver Uninstaller (DDU); NVCleanstall; HWMonitor; fastfetch; bottom; WizTree; Rufus

MSVC runtime (2015+) — cards with checkboxes, any set.

- Tooltip: Official Microsoft packages for the 2015+ (v14) runtime, x86 and x64.
- Note under the field: For a modern system this is the only runtime that matters. For the whole 2005-2013 family use the TechPowerUp or abbodi1406 all-in-one bundle in Classic scripts: it also covers 2012, which WinGet does not have.
- Default: empty
- Options: MSVC 2015+ x86; MSVC 2015+ x64

Advanced: MSVC 2005-2013 — cards with checkboxes, any set.

- Tooltip: Legacy runtimes for old software. WinGet manifests here are frozen at the final vendor build, so they never report an update.
- Note under the field: Official Microsoft packages, one per year and architecture. Recommended instead: the all-in-one bundles from TechPowerUp or abbodi1406 in Classic scripts - one pass, and they include 2012, which has no WinGet package.
- Default: empty
- Options: MSVC 2013 x86; MSVC 2013 x64; MSVC 2010 x86; MSVC 2010 x64; MSVC 2008 x86; MSVC 2008 x64; MSVC 2005 x86

Update available

Load currently available updates from winget upgrade and update only checked items.

Fields

Available updates — cards with checkboxes, any set.

- Default: empty
- The list is built on the fly: `system_core.services.winget_service:available_update_options`

Pin selected

Show pins.txt first, add checked installed IDs to it, and apply blocking WinGet pins.

Fields

pins.txt — cards with checkboxes, any set.

- Note under the field: IDs already listed in config\pins.txt. Keep them at the top and run to apply missing blocking pins.
- Default: empty
- The list is built on the fly: `system_core.services.winget_service:pins_config_options`

Installed packages — cards with checkboxes, any set.

- Note under the field: Select installed IDs to append them to config\pins.txt and apply blocking pins in one run.
- Default: empty
- The list is built on the fly: `system_core.services.winget_service:installed_pin_candidate_options`

Install / uninstall by ID

Search, install typed IDs, and uninstall installed WinGet IDs.

Action buttons

Search

Search the WinGet registry by text.

safe action, no confirmation asked

Search query — text box.

- Default: empty

Add ID to list

Add an exact WinGet ID from search results to a config list and the matching GUI checkbox group.

safe action, no confirmation asked

Package ID — text box.

- Default: empty

Checkbox label — text box.

- Note under the field: Optional. If empty, the ID is used as the label.
- Default: empty

Target list — drop-down list.

- Default: Custom list only
- Options: System; Developer tools; AI; PKMS and notes; Office and documents; Media: images; Media: audio; Media: video; Browsers, VPN, messaging; Hardware and service tools; Custom list only; Pins list only

Install by ID

Install any exact WinGet package ID typed into the field.

changes the system, asks for confirmation before it runs

Package ID — text box.

- Default: empty

Uninstall by ID

Uninstall one exact WinGet package ID typed into the field. Protected system/runtime IDs require an extra checkbox.

changes the system, asks for confirmation before it runs

Package ID — text box.

- Default: empty

Allow protected system/runtime removal — checkbox.

- Note under the field: Required for App Installer, Terminal, PowerShell, MSVC/.NET/WindowsAppRuntime and similar shared dependencies.
- Default: no

Uninstall selected IDs

Load installed WinGet packages into thematic blocks and uninstall checked IDs. Protected system/runtime IDs require an extra checkbox.

changes the system, asks for confirmation before it runs

Allow protected system/runtime removal — checkbox.

- Note under the field: Required for App Installer, Terminal, PowerShell, MSVC/.NET/WindowsAppRuntime and similar shared dependencies. Protected options are labeled in the lists.
- Default: no

System — cards with checkboxes, any set.

- Note under the field: Runtimes, terminals, and archivers. Leave unchecked unless you really mean to remove them.
- Default: empty
- The list is built on the fly: `system_core.services.winget_service:installed_uninstall_system_options`

Developer tools — cards with checkboxes, any set.

- Default: empty
- The list is built on the fly: `system_core.services.winget_service:installed_uninstall_dev_options`

AI — cards with checkboxes, any set.

- Default: empty

- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_ai_options`

PKMS and notes — cards with checkboxes, any set.

- Default: empty
- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_pkms_options`

Office and documents — cards with checkboxes, any set.

- Default: empty
- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_office_options`

Media: images — cards with checkboxes, any set.

- Default: empty
- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_media_images_options`

Media: audio — cards with checkboxes, any set.

- Default: empty
- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_media_audio_options`

Media: video — cards with checkboxes, any set.

- Default: empty
- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_media_video_options`

Browsers, VPN, messaging — cards with checkboxes, any set.

- Default: empty
- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_network_options`

Hardware and service tools — cards with checkboxes, any set.

- Default: empty

- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_hardware_options`

MSVC runtimes — cards with checkboxes, any set.

- Tooltip: One bundle can leave several Programs and Features rows (bundle plus Minimum/Additional Runtime). Check the result with Check MSVC runtimes in Service procedures.
- Note under the field: Every year and architecture in one block on purpose: the clean way to deal with this zoo is to take the whole family out and put it back from an all-in-one bundle in Classic scripts. Visual C++ runtimes are shared dependencies, so applications will break until the bundle is installed.
- Default: empty
- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_msvc_options`

Custom — cards with checkboxes, any set.

- Note under the field: IDs installed through Install by ID or added to config\custom.txt.
- Default: empty
- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_custom_options`

Other installed — cards with checkboxes, any set.

- Note under the field: Installed IDs that are not in project groups or Custom. Review carefully before uninstalling.
- Default: empty
- The list is built on the fly:
  `system_core.services.winget_service:installed_uninstall_other_options`

AI Package Planner

Ask an LLM for package suggestions, validate IDs with WinGet, then execute only selected exact IDs.

Tooltip: AI installer is available only when the selected provider API key is configured.

Action buttons

LLM planning

Provider, model, key and prompt controls for package planning.

Model family — a row of switches, one choice.

- Tooltip: Choose which LLM provider is used for package planning and model cache controls.
- Default: OpenAI
- Options: OpenAI; Gemini

OpenAI API key — drop-down list.

- Tooltip: Select an OpenAI key reference from config or cache. Key material is not written into the GUI cache.
- Default: empty
- The list is built on the fly:
  `system_core.services.winget_ai_service:openai_api_key_options`

OpenAI model — drop-down list.

- Tooltip: Pick a cached or API-discovered OpenAI model for the package plan.
- Default: empty
- The list is built on the fly:
  `system_core.services.winget_ai_service:openai_model_options`

OpenAI model override — text box.

- Tooltip: Optional manual model ID. Use it when the model is valid but not yet present in the cache.
- Default: empty

OpenAI reasoning — a row of switches, one choice.

- Tooltip: Controls reasoning effort for supported OpenAI models: faster answers or deeper planning.
- Default: Fast
- Options: Fast; Balanced; Deep

Gemini API key — drop-down list.

- Tooltip: Select a Gemini key reference from config or cache. Key material is not written into the GUI cache.
- Default: empty
- The list is built on the fly:
  `system_core.services.winget_ai_service:gemini_api_key_options`

Gemini model — drop-down list.

- Tooltip: Pick a cached or API-discovered Gemini model for the package plan.
- Default: empty
- The list is built on the fly: `system_core.services.winget_ai_service:gemini_model_options`

Gemini model override — text box.

- Tooltip: Optional manual Gemini model ID. Use it when the model is valid but not yet present in the cache.
- Default: empty

Instruction template — drop-down list.

- Tooltip: Saved behavior template for the AI planner. It loads text into the AI planner instruction field; selecting it does not scan or install packages.
- Default: empty
- The list is built on the fly: `system_core.services.winget_ai_service:ai_package_prompt_options`

AI planner instruction — markdown_editor.

- Tooltip: Rules for how the AI should reason and format the plan: propose options first, validate WinGet IDs, mention risks, and avoid installing at this step.
- Default: `Подбери Windows/WinGet пакеты под задачу. Сначала предложи варианты, не устанавливай ничего. Отдавай практичный короткий план: что поставить, зачем, какие есть риски или неоднозначности.`

Prompt label — text box.

- Tooltip: Human-readable name used when saving the current prompt to cache.
- Default: empty

Prompt note — text box.

- Tooltip: Optional note for why this prompt exists or what scenario it is tuned for.
- Default: empty

LLM max output — number.

- Tooltip: Upper bound for LLM response size. Increase for large package plans, reduce for short answers.
- Default: `5000`

LLM retries — number.

- Tooltip: How many times to retry a failed LLM request before stopping the operation.
- Default: `2`

LLM timeout — number.

- Tooltip: Maximum wait time for a single LLM request, in seconds.
- Default: `240`

WinGet actions

Search, add and execute exact IDs after review.

No fields: a single button.

Portable Browsers

Download portable browser archives into output and build or update Google Chrome Portable.

Tooltip: Portable mode writes ready archives to the selected output folder and does not install browsers into Windows.

Action buttons

Google Chrome (web installer)

Download the Google Chrome web installer for this system language and architecture, then install it silently.

Tooltip: The ~12 MB web installer picks the locale and the x64/ARM64 build itself at install time, so it always matches the current system. This is a normal Windows install, not a portable build.

changes the system, asks for confirmation before it runs

No fields: a single button.

Google Chrome (download only)

Download the Google Chrome web installer into the portable archives folder without installing it.

Tooltip: Same ~12 MB web installer, saved to output\Portable_archives\ChromeSetup.exe and left there.

safe action, no confirmation asked

No fields: a single button.

Check / install portable 7-Zip

Check Tools\7zip\bin\7za.exe and run install\Install-Portable-7Zip.cmd when it is missing.

changes the system, asks for confirmation before it runs

No fields: a single button.

Download portable browsers

Download selected portable browser packages and place ready ZIP/SFX artifacts into output\Portable.

safe action, no confirmation asked

Portable browsers — cards with checkboxes, any set.

- Tooltip: Choose which portable browser packages to download. Cent Browser is kept as its official portable SFX.
- Default: Ungoogled Chromium Portable, Zen Browser Portable, Cent Browser Portable x64
- At least 1 must be ticked
- Options: Ungoogled Chromium Portable; Zen Browser Portable; Cent Browser Portable x64

Keep temp — checkbox.

- Tooltip: Leave output\Portable_tmp after the operation for inspection or manual packaging checks.
- Default: no

Build Google Chrome Portable

Download Chrome++ and Google Chrome standalone, place Chrome-bin contents into App, and publish output\Portable\Google Chrome Portable. Archive only when enabled.

safe action, no confirmation asked

Chrome source — a row of preset buttons: a press fills the fields below.

- Tooltip: Quickly restore the official Google Chrome standalone x64 installer URL.
- Buttons:
  - **Standalone x64** — Official Google stable standalone installer. Usually this URL does not need to be changed.

Chrome download URL — text box.

- Tooltip: Direct URL to the Google Chrome standalone installer that contains Chrome.7z. Change only when Google moves the payload or you want a custom source.
- Default: `https://dl.google.com/chrome/install/ChromeStandaloneSetup64.exe`

Chrome++ arch — drop-down list.

- Tooltip: Which Chrome++ wrapper folder to use from the Chrome++ archive. The Chrome runtime URL must match the selected architecture.
- Default: x64
- Options: x64; x86; ARM64

Pack archive — checkbox.

- Tooltip: When disabled, publish a ready-to-use Google Chrome Portable folder. When enabled, also package it as ZIP or 7Z.
- Default: no

Archive format — drop-down list.

- Tooltip: Archive format used only when Pack archive is enabled.
- Default: ZIP
- Options: ZIP; 7Z

Keep temp — checkbox.

- Tooltip: Leave output\Portable_tmp after the operation for inspection or manual packaging checks.
- Default: no

Update Google Chrome Portable

Copy Google Chrome Portable from the selected input folder, clear only App, preserve Data and Cache, then publish the updated folder. Archive only when enabled.

safe action, no confirmation asked

Chrome source — a row of preset buttons: a press fills the fields below.

- Tooltip: Quickly restore the official Google Chrome standalone x64 installer URL.
- Buttons:
  - **Standalone x64** — Official Google stable standalone installer. Usually this URL does not need to be changed.

Chrome download URL — text box.

- Tooltip: Direct URL to the Google Chrome standalone installer that contains Chrome.7z. Change only when Google moves the payload or you want a custom source.
- Default: `https://dl.google.com/chrome/install/ChromeStandaloneSetup64.exe`

Chrome++ arch — drop-down list.

- Tooltip: Which Chrome++ wrapper folder to use from the Chrome++ archive. The Chrome runtime URL must match the selected architecture.
- Default: x64
- Options: x64; x86; ARM64

Pack archive — checkbox.

- Tooltip: When disabled, publish a ready-to-use Google Chrome Portable folder. When enabled, also package it as ZIP or 7Z.
- Default: no

Archive format — drop-down list.

- Tooltip: Archive format used only when Pack archive is enabled.
- Default: ZIP
- Options: ZIP; 7Z

Keep temp — checkbox.

- Tooltip: Leave output\Portable_tmp after the operation for inspection or manual packaging checks.
- Default: no

Vendor Downloads

DaVinci Resolve, Affinity, NVIDIA drivers and tools, Windows images with updates inside: every published version straight from the source, each build into its own folder.

Tooltip: WinGet gives the current release; here the vendor's own catalog lists every version. Switch the vendor, pick platform and product, tick the versions or press the buttons on a card. Downloads land in output\Vendors<Vendor><archive name>.

Fields

Vendor — tabs at the top of the form.

- Tooltip: Whose catalog to browse. Windows (UUP dump): every Windows 11 and 10 build with its cumulative update, as a script package that builds the ISO from Microsoft's own files. TechPowerUp: AMD, Intel and Qualcomm drivers plus the site's utilities, monitors, benchmarks and video BIOS flashers, every version. NVIDIA: Game Ready and Studio drivers by chip generation, NVIDIA App, Broadcast, CUDA Toolkit and DLSS. Blackmagic: Resolve, Fusion, RAW, Desktop Video and the whole catalog. Affinity: Photo, Designer, Publisher 2 and the unified Affinity 3.
- Default: Windows (UUP dump)
- Options: Windows (UUP dump); TechPowerUp — TechPowerUp's catalog with the whole version history of every entry: AMD, Intel and Qualcomm drivers, tuning utilities (DDU, NVCleanstall, ThrottleStop...), monitors (GPU-Z, CPU-Z, AIDA64), benchmarks and video BIOS flashers. NVIDIA drivers and DLSS stay on the NVIDIA tab.; NVIDIA; Blackmagic Design; Affinity

Section — a row of switches, one choice.

- Tooltip: The section of TechPowerUp's catalog: drivers, tuning utilities, monitoring, benchmarks or video BIOS flashers. The product row below changes with it. NVIDIA drivers and the DLSS libraries stay on the NVIDIA tab.
- Default: Drivers
- Shown when: Vendor = 'TechPowerUp'
- Options: Drivers; Utilities; Monitoring; Benchmarks; Video BIOS

Product — a row of switches, one choice.

- Tooltip: Drivers TechPowerUp mirrors with their whole version history, the same installers the vendors ship. Handy when Intel or AMD hide the older ones on their own sites. TechPowerUp keeps every version with date, size and checksum; the file comes from TechPowerUp's own signed link.
- Default: AMD Radeon Graphics
- Shown when: Vendor = 'TechPowerUp'; Section = 'Drivers'

- Options: AMD Radeon Graphics; AMD Ryzen Chipset; Intel Graphics (Arc, iGPU); Intel Wi-Fi; Intel Bluetooth; Intel Ethernet; Intel NPU; Qualcomm Snapdragon X Graphics

Product — a row of switches, one choice.

- Tooltip: Tuning and clean-up tools: DDU removes a display driver completely, NVCleanstall installs an NVIDIA driver without the extras, Profile Inspector edits driver profiles, ThrottleStop tames laptop CPU throttling, DRAM Calculator and Samsung Magician serve memory and SSDs, the Visual C++ package installs every runtime at once, MemTest64 checks RAM from Windows. TechPowerUp keeps every version with date, size and checksum; the file comes from TechPowerUp's own signed link.
- Default: DDU
- Shown when: Vendor = 'TechPowerUp'; Section = 'Utilities'
- Options: DDU; NVCleanstall; NVIDIA Profile Inspector; ThrottleStop; DRAM Calculator for Ryzen; Samsung Magician; Visual C++ Runtimes AIO; MemTest64

Product — a row of switches, one choice.

- Tooltip: Sensors and system information: GPU-Z and CPU-Z read the chips, AIDA64 covers the whole machine, ZenTimings shows Ryzen memory timings, Real Temp reads core temperatures. TechPowerUp keeps every version with date, size and checksum; the file comes from TechPowerUp's own signed link.
- Default: GPU-Z
- Shown when: Vendor = 'TechPowerUp'; Section = 'Monitoring'
- Options: GPU-Z; CPU-Z; AIDA64 Extreme; ZenTimings; Real Temp

Product — a row of switches, one choice.

- Tooltip: Load and score tools: 3DMark and PCMark, Cinebench for the CPU, FurMark for the GPU, Prime95 and Linpack for stability, the Unigine scenes and GravityMark for graphics, ATTO for disks. TechPowerUp keeps every version with date, size and checksum; the file comes from TechPowerUp's own signed link.
- Default: 3DMark
- Shown when: Vendor = 'TechPowerUp'; Section = 'Benchmarks'
- Options: 3DMark; PCMark 10; Cinebench; FurMark; Prime95; Unigine Superposition; Unigine Heaven; GravityMark; Linpack Xtreme; ATTO Disk Benchmark

Product — a row of switches, one choice.

- Tooltip: Video BIOS flashers: NVFlash for NVIDIA cards, AMDVBFlash for AMD cards. Every version, back to the ones old cards still need. TechPowerUp keeps every version with date, size and checksum; the file comes from TechPowerUp's own signed link.
- Default: NVIDIA NVFlash
- Shown when: Vendor = 'TechPowerUp'; Section = 'Video BIOS'
- Options: NVIDIA NVFlash; AMDVBFlash

Windows page — tabs at the top of the form.

- Tooltip: The Windows tab in three pages. Version: product, platform, edition, language, image kind and the build cards, the newest of each generation in gold. Apps: which Store apps go into the image. Drivers: this machine's drivers to export and embed. The options row on top applies to all three.
- Default: Version
- Shown when: Vendor = 'Windows (UUP dump)'
- Options: Version — Which Windows: product, platform, edition and language pick the list; every build of the catalog with its cumulative update is a card, tick the one to build.; Apps — Which Store apps go into the image: the sets, the group buttons, the cards.; Drivers — This machine's drivers: tick, export to output, embed from input.

Platform — a row of switches, one choice.

- Tooltip: Which build of the Blackmagic catalog to list. Windows ARM is the Snapdragon build.
- Default: Windows
- Shown when: Vendor = 'Blackmagic Design'
- Options: Windows; Windows ARM; macOS; Linux

Product — a row of switches, one choice.

- Tooltip: The frequent Blackmagic products by name; 'Whole catalog' lists every product for the platform with the product in front of each version.
- Default: Resolve Studio
- Shown when: Vendor = 'Blackmagic Design'
- Options: Resolve Studio; Resolve (free); Fusion Studio; Fusion; Blackmagic RAW; Desktop Video; Camera Setup; Whole catalog

Platform — a row of switches, one choice.

- Tooltip: Windows is the x64 build for Intel and AMD, Windows ARM the arm64 build for Snapdragon, macOS the dmg.
- Default: Windows
- Shown when: Vendor = 'Affinity'
- Options: Windows; Windows ARM; macOS

Product — a row of switches, one choice.

- Tooltip: Affinity 3 is the unified free app from Canva, current release only. The version 2 apps come from the Serif updates pages with every 2.x release, as msix and exe on Windows.
- Default: Affinity 3
- Shown when: Vendor = 'Affinity'
- Options: Affinity 3; Photo 2; Designer 2; Publisher 2; All four

Product — a row of switches, one choice.

- Tooltip: Drivers by chip generation, or one of the tools. Every driver version is marked with the NVENC SDK generation it satisfies, the newest FFmpeg build from Audion Media Tools that encodes on it, the newest CUDA Toolkit it runs, and a star on the keepers; the marks come from config\vendor_nvidia.yaml. DLSS is not a program but the DLL games carry: every version of the three libraries comes from TechPowerUp, to drop into a game folder or feed to the NVIDIA App override.
- Default: Game Ready
- Shown when: Vendor = 'NVIDIA'
- Options: Game Ready; Studio; NVIDIA App; Broadcast; CUDA Toolkit; DLSS; DLSS Frame Gen; DLSS Ray Rec

Chip generation — a row of switches, one choice.

- Tooltip: The generation the driver must support. One driver file usually covers several generations; the list shows what NVIDIA publishes for the chosen one, so the oldest version here is the oldest that still runs those chips. 'My cards' keeps only the versions published for every generation named in config\vendor_nvidia.yaml. GTX 700 and 600 end with the 47x branch, GTX 500 and 400 with the last driver 391.35: NVIDIA files it under Windows 10, it installs on Windows 11 all the same.
- Default: My cards
- Shown when: Vendor = 'NVIDIA'; Product = 'Game Ready' / 'Studio'

- Options: My cards; RTX 50; RTX 40; RTX 30; RTX 20; GTX 16; GTX 10; GTX 900; GTX 700; GTX 600; GTX 500; GTX 400

My generations — cards with checkboxes, any set.

- Tooltip: The chip generations in your own machines. 'My cards' in the generation row shows only the driver versions NVIDIA published for every ticked generation, so one file serves them all. The ticks are remembered in config\vendor_nvidia.yaml and come back on the next start.
- Default: empty
- The list is built on the fly: `system_core.services.vendor_service:nvidia_my_generation_options`
- Shown when: Vendor = 'NVIDIA'; Chip generation = 'My cards'

Form — a row of switches, one choice.

- Tooltip: Desktop cards or notebook GPUs: NVIDIA keeps separate driver lists for the two.
- Default: Desktop
- Shown when: Vendor = 'NVIDIA'; Product = 'Game Ready' / 'Studio'
- Options: Desktop; Notebook

Product — a row of switches, one choice.

- Tooltip: Every build the catalog has seen, each cumulative update as its own build number with its date, newest first. Type a release like 24H2 or a month into the filter to narrow the list.
- Default: Windows 11
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Version'
- Options: Windows 11; Windows 10; Insider

Platform — a row of switches, one choice.

- Tooltip: amd64 for Intel and AMD machines, arm64 for Snapdragon. The catalog keeps both.
- Default: Windows x64
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Version'
- Options: Windows x64; Windows ARM

Edition — a row of switches, one choice.

- Tooltip: Home or Pro, or both in one image; the N editions come without the media components.
- Default: Pro
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Version'

- Options: Pro; Home; Home + Pro; Pro N; Home N

Language — a row of switches, one choice.

- Tooltip: The language of the image. Two buttons for the usual choice; 'Other' opens the list of the remaining 36 languages the UUP dump catalog offers. The default follows the interface language: English gives en-us, Russian ru-ru.
- Default: English (US)
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Version'
- Options: English (US); Русский; Other...

Other language — drop-down list.

- Tooltip: The remaining 36 languages of the UUP dump catalog. Type to search.
- Default: English (UK, en-gb)
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Version'; Language = 'Other...'
- Options: العربية (ar-sa); Български (bg-bg); Čeština (cs-cz); Dansk (da-dk); Deutsch (de-de); Ελληνικά (el-gr); English (UK, en-gb); Español (es-es); Español (México, es-mx); Eesti (et-ee); Suomi (fi-fi); Français (Canada, fr-ca); Français (fr-fr); עברית (he-il); Hrvatski (hr-hr); Magyar (hu-hu); Italiano (it-it); 日本語 (ja-jp); 한국어 (ko-kr); Lietuvių (lt-lt); Latviešu (lv-lv); Norsk (nb-no); Nederlands (nl-nl); Polski (pl-pl); Português (Brasil, pt-br); Português (pt-pt); Română (ro-ro); Slovenčina (sk-sk); Slovenščina (sl-si); Srpski (sr-latn-rs); Svenska (sv-se); ไทย (th-th); Türkçe (tr-tr); Українська (uk-ua); 中文（简体, zh-cn）; 中文（繁體, zh-tw)

Image kind — a row of preset buttons: a press fills the fields below.

- Tooltip: One press sets the base edition and the additional ones the way Microsoft's own media do it. Business: Pro with Enterprise, Education, Pro Education and Pro for Workstations. Consumer: Home and Pro with Education, Pro Education and Pro for Workstations. Pro only: Pro alone.
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Version'
- Buttons:
  - **Business** — Pro + Enterprise, Education, Pro Education, Pro for Workstations - what the business editions ISO carries.
  - **Consumer** — Home and Pro + Education, Pro Education, Pro for Workstations - what the consumer editions ISO carries.
  - **Pro only** — Pro alone, nothing added.

Additional editions in the ISO — cards with checkboxes, any set.

- Tooltip: Editions UUP dump derives from Pro while building: they are added to the same ISO next to the base edition. LTSC is not among them, Microsoft does not publish it this way. Leave everything off for a plain Home or Pro image.
- Default: empty
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Version'
- Options: Enterprise; IoT Enterprise; IoT Enterprise K; Education; Pro Education; Pro for Workstations; Enterprise multi-session

Built-in apps — a row of switches, one choice.

- Tooltip: Which Store apps the image carries. 'As Microsoft' keeps the stock set. 'Chosen below' builds the image with the ticked apps only, the converter supports this since build 22563. 'None' adds no Store apps at all, not even the Store itself.
- Default: Chosen below
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Apps'
- Options: As Microsoft; Chosen below; None

App set — a row of preset buttons: a press fills the fields below.

- Tooltip: One press ticks a set below. Minimal: the four system packages, Store, Store purchases, Windows Security and App Installer. Work: those plus thirteen tools and ten codecs, 27 packages, the default and the set the reference image was built with. Everything: the whole catalog. Every card can be changed afterwards.
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Apps'; Built-in apps = 'Chosen below'
- Buttons:
  - **Minimal**
  - **Work**
  - **Everything**

Include in the distribution build — group toggle buttons.

- Tooltip: Groups of built-in apps. A button that is not pressed keeps its group out of the image; press it and the group is installed. The number is how many packages the group holds, the list is in the button's own tooltip. Works for every Windows 11 edition, not only N. The cards below show the same choice package by package.

- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Apps'; Built-in apps = 'Chosen below'
- Buttons:
  - **Media stack (4)** — Media Player, Films & TV, Photos, Clipchamp. OneDrive and Copilot are not Store packages and are removed after installation.
  - **Edge (1)** — The Edge browser, the converter's SkipEdge option in reverse. The Edge WebView2 runtime is untouched and stays in the system: it is a separate component, apps run and are built on it, from the new Outlook and Teams to third-party programs.
  - **Xbox (6)** — Xbox app, Xbox Game Bar, Game overlay, speech-to-text overlay, Xbox identity provider, Xbox TCUI.
  - **Teams and mail (6)** — Teams, Outlook (new), Mail and Calendar, People, Office hub (M365), Family.
  - **Bing and widgets (6)** — Bing Search, News, Widgets (web experience), Widgets runtime, Start experiences, Cortana.
  - **Promo and helpers (8)** — Solitaire, Tips, Get Help, Feedback Hub, Quick Assist, Dev Home, PC Manager, App compatibility enhancements.
  - **Small tools (2)** — Maps, Cross Device. Clock, Sticky Notes and Phone Link belong to the Work set and follow the cards below.

Apps in the image — cards with checkboxes, any set.

- Tooltip: Ticked apps go into the image, the rest are never installed, so there is nothing to remove afterwards. The Store, Windows Security and App Installer are worth keeping: WinGet and Store updates run through them. The codec extensions give Explorer and the Photos-free system HEIF, WebP, HEVC and AV1 thumbnails and playback.
- Default: 27: Microsoft Store, Store purchases, Windows Security, App Installer (WinGet), Notepad, Windows Terminal, Calculator, Snipping Tool, Paint, Camera, Sound Recorder, Phone Link ...
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Apps'; Built-in apps = 'Chosen below'
- Options (59 items): Microsoft Store; Store purchases; Windows Security; App Installer (WinGet); Notepad; Windows Terminal; Calculator; Snipping Tool; Paint; Camera; Photos; Clock; Sticky Notes; Maps; Sound Recorder; Media Player; Films & TV; Clipchamp; Phone Link; Cross Device; Mail and Calendar; Outlook (new); People; Teams; To Do; Office hub (M365); Cortana; Bing Search; News; Weather; Widgets (web experience); Widgets runtime; Start experiences; Xbox app; Xbox Game Bar; Xbox Game overlay; Xbox speech overlay; Xbox identity; Xbox TCUI; Solitaire; Feedback Hub; Get Help; Tips; Quick Assist; Family; Power Automate; Dev Home; PC Manager; App compatibility

enhancements; Codec: Web Media; Codec: RAW images; Codec: HEIF; Codec: HEVC; Codec: VP9; Codec: WebP; Codec: AV1; Codec: MPEG-2; Codec: AVC encoder; Codec: Dolby Audio

This machine's drivers: tick, press Export drivers, the packages land in output\Drivers — cards with checkboxes, any set.

- Tooltip: Every third-party driver package of this machine (in-box Microsoft ones are not listed). Class, original inf name, provider and version on each card. The network classes are ticked when the list opens, only the newest version of each driver; 'Select block' and 'Clear block' take the whole list.
- Default: empty
- The list is built on the fly: `system_core.services.vendor_service:machine_driver_options`
- At least 1 must be ticked
- Shown when: Vendor = 'Windows (UUP dump)'; Windows page = 'Drivers'

Pick — a row of preset buttons: a press fills the fields below.

- Tooltip: Ticks version cards by rule instead of by hand. Latest stable: the newest final release, betas and Insider builds skipped. Portable: only the portable builds and zips stay in the list, press again for the installers. Golden: every starred card. Clear: nothing ticked.
- Buttons:
  - **Latest stable** — The newest final release; betas, previews and Insider builds are skipped.
  - **Portable** — Portable mode: the list keeps only the versions that have a portable build or zip, and a tick now means that file, the one behind the crimson archive icon. Ticks you already made move to the portable file of the same version. Press again for the installers. An empty list means the product has no portable build. (shown when: Vendor = 'TechPowerUp')
  - **★ Golden** — Every starred card. NVIDIA: the drivers marked golden in config\vendor_nvidia.yaml. Windows: the newest build of each generation, 22H2, 24H2, 25H2 and so on, with its latest cumulative update; the rest of a generation is history. (shown when: Vendor = 'NVIDIA' / 'Windows (UUP dump)')
  - **Clear** — No card ticked.

Portable builds only — checkbox.

- Hidden field: its value goes to the operation, another control drives it on screen.
- Tooltip: The Portable button of the pick row: while on, the version list keeps only versions with a portable build and a tick means that file. Counts on the TechPowerUp tab only.
- Default: no

- Shown when: Vendor = 'TechPowerUp'

Versions — cards with checkboxes, any set.

- Tooltip: Every version the vendor has for this product and platform, newest first. Tick the ones to fetch, or use the buttons on a card for that one build; the filter box narrows the list. On a card the green arrow fetches the installer, the crimson archive icon fetches the portable build or zip of the same version, the blue link only prints the download link; an empty slot means the version has no such file.
- Default: empty
- The list is built on the fly: `system_core.services.vendor_service:vendor_build_options`
- At least 1 must be ticked

Unpack zip — checkbox.

- Tooltip: Unpack the archive into its folder, so the installer is at hand. When everything in the zip sits in one root folder, that folder is dropped and the files land at the top. Only zip; exe, msix, dmg and tar stay as they are.
- Default: yes

Delete zip after unpacking — checkbox.

- Tooltip: Once unpacked, the archive only doubles the disk use. Off keeps both.
- Default: yes

Without Edge — checkbox.

- Hidden field: its value goes to the operation, another control drives it on screen.
- Tooltip: Tell the converter to leave the Edge browser out of the image (its SkipEdge option). The Edge WebView2 runtime is untouched and stays in the system: it is a separate component, apps run and are built on it, from the new Outlook and Teams to third-party programs.
- Default: yes
- Shown when: Vendor = 'Windows (UUP dump)'

Build the ISO right away — checkbox.

- Tooltip: After the package is unpacked, start its uup_download_windows.cmd in a console: it downloads the files from Microsoft with aria2 and assembles the ISO. The build runs in the UUP folder at the root of the program's drive; the finished image is moved to the destination folder. Needs administrator rights; takes a while.

- Default: yes
- Shown when: Vendor = 'Windows (UUP dump)'

Delete cache after ISO — checkbox.

- Tooltip: Once the image is moved to the destination, the build folder in the UUP folder at the root of the program's drive, with the converter's scripts and the 9 GB of Microsoft's files, is deleted. Off: the folder stays, so the same build can be reassembled with other apps without downloading again.
- Default: no
- Shown when: Vendor = 'Windows (UUP dump)'; Build the ISO right away = 'True'

Embed drivers from input into the image — checkbox.

- Tooltip: Step by step. 1) Get driver packages: open 'This machine's drivers' in the operations list, the network drivers are ticked already, tick more if you want, press Export: the packages go into output\Drivers as folders with .inf, .sys and .cat; or copy such folders from anywhere. 2) Put them anywhere inside the input folder (the Source path at the top of the window): the exported Drivers folder as is, a folder named after the laptop, or the packages at the top, the program finds every .inf itself. 3) Tick this box and build the ISO: each package is copied into the build's Drivers\OS and DISM embeds them into install.wim, so Windows Setup installs them with the system. Only unpacked INF packages work, a vendor's .exe installer does not; the architecture must match the image (x64); unsigned drivers need test signing on the target. Storage or RAID drivers that Setup itself needs to see the disk are not covered here.
- Default: no
- Shown when: Vendor = 'Windows (UUP dump)'

Straight into the destination folder — checkbox.

- Tooltip: Skip the Vendors<Vendor> levels: each build folder lands directly in the destination. For a store that already is the vendor's folder.
- Default: no

First name — text box.

- Tooltip: Only for the free DaVinci Resolve: Blackmagic issues its link after this form. Sent straight to blackmagicdesign.com, nothing is kept here.
- Default: empty
- Shown when: Vendor = 'Blackmagic Design'; Product = 'Resolve (free)' / 'Whole catalog'

Last name — text box.

- Tooltip: Part of the Blackmagic form for the free Resolve.
- Default: empty
- Shown when: Vendor = 'Blackmagic Design'; Product = 'Resolve (free)' / 'Whole catalog'

E-mail — text box.

- Tooltip: Part of the Blackmagic form for the free Resolve. No confirmation letter is sent.
- Default: empty
- Shown when: Vendor = 'Blackmagic Design'; Product = 'Resolve (free)' / 'Whole catalog'

Phone — text box.

- Tooltip: Part of the Blackmagic form for the free Resolve.
- Default: empty
- Shown when: Vendor = 'Blackmagic Design'; Product = 'Resolve (free)' / 'Whole catalog'

Country code — text box.

- Tooltip: Two-letter code as the Blackmagic site uses it, for example us or de.
- Default: empty
- Shown when: Vendor = 'Blackmagic Design'; Product = 'Resolve (free)' / 'Whole catalog'

City — text box.

- Tooltip: Part of the Blackmagic form for the free Resolve.
- Default: empty
- Shown when: Vendor = 'Blackmagic Design'; Product = 'Resolve (free)' / 'Whole catalog'

Street — text box.

- Tooltip: Part of the Blackmagic form for the free Resolve.
- Default: empty
- Shown when: Vendor = 'Blackmagic Design'; Product = 'Resolve (free)' / 'Whole catalog'

State / region — text box.

- Tooltip: Part of the Blackmagic form for the free Resolve.
- Default: empty
- Shown when: Vendor = 'Blackmagic Design'; Product = 'Resolve (free)' / 'Whole catalog'

Action buttons

Download

Fetch the ticked versions into a folder per build, resuming an interrupted file, and unpack the zip when asked.

safe action, no confirmation asked

Link only

Resolve the download links for the ticked versions and print them to the log and to report\links.txt; nothing is downloaded.

safe action, no confirmation asked

Install ADK

Why: the UUP dump converter puts the cumulative update, the extra editions and the drivers into the image with DISM, and it takes that DISM from the Windows ADK Deployment Tools; the system DISM of a newer host refuses, and the image comes out without updates. This button fetches Microsoft's adksetup.exe and installs only the Deployment Tools, about 100 MB, once per machine. Asks for administrator rights. Until it is installed, the Windows download stays closed.

changes the system, asks for confirmation before it runs; shown when: vendor = uupdump

WinUtil

Open Chris Titus's Windows utility (irm https://christitus.com/win | iex) in its own terminal: tweaks, debloat, program installs, Windows updates control. Runs in Windows Terminal when it is installed, otherwise in a plain console, with PowerShell 7 when present and Windows PowerShell 5.1 as the fallback. The script is downloaded from the author's site at launch and asks for administrator rights in its own window.

changes the system, asks for confirmation before it runs; shown when: vendor = uupdump

Export drivers

Step by step. 1) In the section This machine's drivers below, the network drivers are ticked already; tick anything else you want. 2) Press this button: pnputil /export-driver copies every ticked driver into output\Drivers, one folder per package with .inf, .sys and .cat; nothing on the machine changes. 3) Move that Drivers folder into the input folder (the Source path at the top of the window), under any

name. 4) Tick 'Embed drivers from input into the image' in the Options row at the top of this tab together with 'Build the ISO right away': while building, the program finds every .inf anywhere under input and DISM embeds the packages into install.wim, so Windows Setup installs them with the system. Without that checkbox the drivers stay in input and are not embedded.

safe action, no confirmation asked; shown when: vendor = uupdump

Import / export

WinGet JSON import and export with GUI fields.

Action buttons

Export JSON

Export installed WinGet packages to output.

safe action, no confirmation asked

Export path — text box.

- Default: `output\winget-export.json`

Include versions — checkbox.

- Default: yes

Import JSON

Import packages from a WinGet export file.

changes the system, asks for confirmation before it runs

Import path — text box.

- Default: `input\winget-export.json`

Ignore versions — checkbox.

- Default: no

Do not upgrade installed packages — checkbox.

- Default: no

Ignore unavailable packages — checkbox.

- Default: yes

Classic scripts

Compatibility commands that run existing project CMD scripts.

Action buttons

Update WinGet (script)

Run the existing Microsoft App Installer update script.

changes the system, asks for confirmation before it runs

No fields: a single button.

MSVC All-in-One (TechPowerUp)

Download the Visual C++ Runtimes All-in-One package and unpack it; nothing is installed.

Tooltip: One archive with every VC++ runtime from 2005 to 2015+, x86 and x64, plus its own install_all.bat. Resolved through the TechPowerUp mirror flow at click time because the file URL is signed and expires.

safe action, no confirmation asked

No fields: a single button.

MSVC AIO (abbodi1406)

Download the latest VisualCppRedist_AIO x86+x64 release; nothing is installed.

Tooltip: One self-contained executable that installs every VC++ runtime from 2005 to v14, including 2012, which WinGet does not carry. Silent switch is /ai and it needs elevation, so Audion Get Tools only downloads it.

safe action, no confirmation asked

No fields: a single button.

Install MSVC All-in-One (TechPowerUp)

Download the TechPowerUp bundle and run install_all.bat: every VC++ runtime from 2005 to v14, x86 and x64.

Tooltip: A third-party script installs ten vendor packages one after another with /passive /norestart. Administrator rights are required; progress windows are normal. The registry state before and after is written to the log.

changes the system, asks for confirmation before it runs

No fields: a single button.

Install MSVC AIO (abbodi1406)

Download VisualCppRedist_AIO and run it silently with /ai: every VC++ runtime including 2012.

Tooltip: One third-party executable installs the whole family without questions. Administrator rights are required. The registry state before and after is written to the log, because the installer's own exit code says little.

changes the system, asks for confirmation before it runs

No fields: a single button.

Install MSVC 2015+

Run Install-Audion-MSVC-2015+.cmd.

changes the system, asks for confirmation before it runs

No fields: a single button.

Install MSVC Legacy

Run Install-Audion-MSVC.cmd.

changes the system, asks for confirmation before it runs

No fields: a single button.

Update MSVC 2015+

Run Update-Audion-MSVC-2015+.cmd.

changes the system, asks for confirmation before it runs

No fields: a single button.

Service procedures

Buttons of the Maintenance section. Each runs at once, without a form.

Windows licence state

Read-only: edition, licence channel and expiry of this Windows through slmgr, printed to the log.

safe action, no confirmation asked

Health / Doctor

Check WinGet availability, sources, installed/update counts, and run the GUI doctor.

safe action, no confirmation asked

Check MSVC runtimes

Read the real Visual C++ runtime versions from the registry, compare the 2015+ family with WinGet, and list every ARP entry.

Tooltip: Only VC\Runtimes<arch> is trusted for versions; the ARP display name has been renamed twice and one bundle leaves several rows. 2012 has no WinGet package and is reported honestly as such.

safe action, no confirmation asked

Check installed IDs

Diagnostic check: run winget list for checked package IDs without installing or updating them.

safe action, no confirmation asked

Validate lists

Count package lines and report duplicates in config lists.

safe action, no confirmation asked

Clear I/O

Delete files inside managed input and output folders.

changes the system, asks for confirmation before it runs

Clear logs

Delete old files inside the managed logs folder while preserving the current operation log.

changes the system, asks for confirmation before it runs