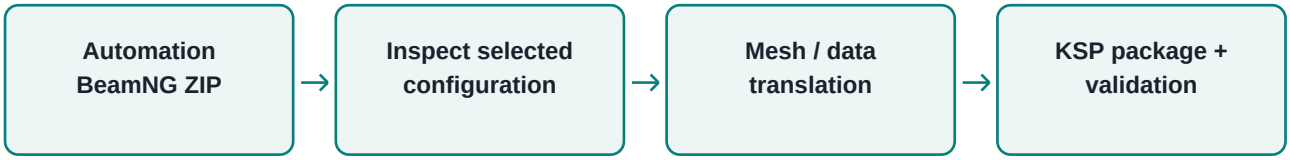
Runtime kda25 / Converter V2.0 kda12 | Updated 2026-10-28

01 Offline workflow and inputs

KDA Vehicle Converter V2.0, build 2026-10-26-kda12. This version performs conversion locally without AI, tokens or an internet connection. It uses source/converter/kdaconvert with bundled Python/native dependencies and requires a compatible base (minimum kda14); install current kda25 for these manuals.



Recommended export

Use Automation's BeamNG exporter with ZIP packing enabled. Disable Merge All Fixtures to preserve separate trim, grille, lamps, canards and exhaust components. Merged fixtures also work but reduce component-level identification. Collapse Livery may simplify texture layers. High-resolution input is optional because the converter has its own output quality choices.

A native .car file carries useful engine/family metadata but does not contain the complete ready-to-convert visual model. The supported input is the BeamNG ZIP, which may already embed the .car. A standalone .car is deliberately rejected.

Desktop controls

Select export ZIP, output parent folder, texture quality, KSP folder and optional save. Convert only writes a validated package; Convert and install also copies it to KSP. Install last conversion retries a ready package later. High / Balanced / Light cap texture dimensions at 2048 / 1024 / 512 respectively.

Planned seats is metadata only. The generated craft has five chassis parts. Add physical KDA race seats and doorway markers in SPH, then assign Kerbals. The separately installed KDA crew addon removes hidden body command/crew slots. The supplied Lizard demo craft already has one positioned seat and marker.

02 Import data and provenance

Input	Translation
Selected .pc / JBeam slots	Resolves the actual engine, transmission, drive split, aero, tires and fixtures selected by that car configuration.
Torque / boost curves	Imports rpm-dependent torque; separates boosted calibration to avoid double-counting; imports reference boost and approximated transient spool.
Embedded .car	Reads engine layout, name, bore, stroke and displacement when available. Audio names are a marked fallback for layout, not a displacement estimate.
Gearbox / differential	Forward and reverse ratios, final drive, locked imported gearbox type, drive axle weights and configurable runtime LSD behavior.
Tires	Corner position, radius, width, compound/friction calibration and normalized driven-wheel shares.
Aero	Signed front/rear cefaero forces, reference speed/density, points, drag area and axis multipliers.
Audio	Original shared coast/load banks at eight RPM anchors; live intake/boost/shift detail. Four combustion layers per audible car and four audible cars maximum. No exported engine WAVs copied.

vehicle.json stores the input ZIP SHA-256, selected configuration, imported values, warnings, visual translations and runtime part IDs. converter version/build and required framework ABI/minimum build are recorded. Do not hide a fallback behind a plausible-looking value; warnings identify missing or approximate data.

Current adapter targets four-wheel Automation vehicles and supported engine/gearbox schemas. Unsupported source structures should produce useful failures or retain rigid art; arbitrary BeamNG native vehicles are not guaranteed. Audio and licensing are separate concerns: technical access to a WAV is not permission to publish it.

03 Meshes, materials and articulation

Independent wheel configuration

The converter writes each wheel configuration directly from the KDA wheel contract, without loading a external CFG template. It preserves tested mass, grip, suspension, geometry, brakes, drive shares and module order. A neutral native motor curve supplies setup normalization; the vehicle engine controls actual torque. Stock automatic steering reduction remains disabled.

Geometry pipeline

COLLADA geometry and selected fixtures become KSP .mu models through the vendored MU writer. Coordinates, outward rim orientation and wheel-centered rotation pivots are normalized for the existing wheel rig. Meshes are split below the 65,535 vertex limit. Models are rigid; JBeam deformation and BeamNG Lua are not imported.

Body and wheels are independent parts. Wheel selection box colliders make them selectable after attachment. Cosmetic meshes do not add independent masses or physics joints. Underframe, engine and exhaust are retained as rigid visual geometry.

Paint and transparency

Albedo/livery/fixture masks preserve holes and trim overlays. Roughness, metalness and normals are translated to a stock KSP fallback and optional Resurfaced PBR definitions. Material category heuristics reduce over-shiny cast components. Multiple surfaces must not collapse into one opaque body material that covers exported cutouts.

External PBR mods are not included. Download HarmonyKSP, Shabby and Resurfaced separately using the main README. ModuleManager is required for KDA part patches. The converter writes PBR definitions but does not install those external mods. Add converted cars beneath KDAVehicles/Vehicles after installing KDA and the external dependencies.

Suspension and exhaust

Supported independent suspension arms use JBeam fixed/moving anchors and native wheel travel for a cosmetic rig. The converter retains unsupported live-axle/missing-anchor hardware rigidly and records the reason. Exhaust tips are detected and generate integrated effect outlets; multiple outlets are supported. Detection remains heuristic and should be checked visually.

Interior shell rendering is supplied by the crew runtime, not by generating a conventional bespoke cockpit IVA for every exported vehicle. Source geometry and masks determine the available detail.

04 Output, installation and safeguards

Output path	Contents
GameData/KDAVehicles/Vehicles/<id>	Parts/body.cfg, four corner cfg files, five MU models, textures, Sounds, PBR definitions, vehicle.json and source license.
Ships/SPH/<id>.craft	Assembled body and four wheels; manual seats/doorways still required.
validation.json / desktop-session.log	Package checks and export warnings. Not a certification of live driving behavior.

Conversion rejects path traversal/symlinks and stages output before a final move. Existing output folders are not overwritten. Validation checks required model/part files, texture references, finite mesh geometry, selection collider structure, audio format and runtime consistency.

Installer refuses a running KSP, duplicate runtimes, invalid base ABI/build/hash, missing crew files, conflicting part IDs and redirected paths. It installs only the car and craft, with scoped backups and rollback. Base runtime, crew and garage files are installed separately and are never replaced by the converter.

Selecting a save writes one generated craft there and archives an older stock copy to avoid duplicate load-menu entries. It does not delete unrelated craft or remove every older manually imported vehicle.

Manual copy installation must likewise preserve one runtime and dependency copy. Do not paste multiple old converter packages over each other without examining version/hash and vehicle IDs.

05 Build and verification

Rebuild from source

Use Windows, Python 3.12, NumPy/Pillow and PyInstaller. From the repository's source directory, install converter/requirements.txt and PyInstaller, then run tools/build_converter.ps1 -Python <python.exe> -OutputDirectory ../Converter. The vendored MU writer and conversion dependencies are included; no framework/garage payload or Blender installation is required.

GameData is the only release asset tree. Runtime/crew builds stage into source/.build/runtime. build_plugin.ps1 and build_crew_plugin.ps1 require -KspRoot pointing to a legally installed KSP. KSP/Unity assemblies are external and are not redistributed. The converter build does not use a runtime-resource snapshot.

Headless conversion

KDAVehicleConverter.exe --convert "export.zip" --output "new-package-folder" --quality balanced. Optional --ksp and --save request installation. Native CLI inspect-only is available through python -m kdaconvert for metadata investigation; the desktop entry point is the recommended conversion/install path.

Release verification

Check GUI build text, actual runtime SHA and fresh package validation. Inspect rim faces, spinning pivots, alpha holes, paint/metal roughness, underbody/exhaust and rear suspension warnings in native KSP. Run installer failure tests against disposable directories. Confirm no second runtime and no personal telemetry/save files are in the release.

The source includes targeted parser/material/installation tests. Some tests require external car ZIPs and historical fixtures and are not a self-contained universal test suite. The repository contains Lizard GT and Rabbit Rally x as playable cars; source testing support does not add other car parts to GameData.

This version includes original hybrid engine audio with shared generated banks and bounded playback. Standard/Rally/Desert travel remains implemented; a direct Automation plugin is future work. Original audio and manual text are CC BY 4.0; KDA/Sparky branding has separate limited permission in ASSET-LICENSES.md. Third-party vehicle/artwork rights require their own review.

06 Current base mod and garage

Install the current base GameData separately. The converter produces only a car folder, craft and validation/metadata files. It neither bundles nor upgrades runtime DLLs, crew resources, garage assets or dependency mods.

The base runtime provides Standard/Rally/Desert suspension, axle travel/height/track tuning, compression/rebound and roll graphs, independent tire grip, brake power, driving modes, AWD hardware behavior and telemetry. Existing cars use their saved settings until edited.

Garage Cars / Build / Tune includes camera views, continuous loading, confirmed car deletion, protected input, physical seat/door placement, exhaust tunes and opt-in rechargeable nitrous. The flight HUD can be reopened through its KDA mod-toolbar button.

The optional world garage uses original procedural KDA artwork, catalog KDAGarage and model Garage/garage.mu. Install Kerbal Konstructs and Custom Pre Launch Checks separately. A named KDAEntrance controls the front-only F8 zone and door-facing driveway launch. An optional default KSC placement is included under Garage/Garage-instances.cfg. It uses KSC_Builtin on Kerbin, position -396.634735,24.5667,372.811676, orientation 0,150.000214,0 and scale 2. Press Ctrl+K in KK to move it; omit this instance file to retain an existing manual placement without a duplicate. Player sessions are not packaged.

The Python CLI and desktop app share conversion/installation services. Output folders must be new. Imported fixtures, articulation and materials have limits documented in Converter.md. Current package hashes and completed check scope are in release-manifest.json and documentation/release-manifest.json.

07 Version, credits and licenses

Current runtime: V1.0 / 2026-10-28-kda25. Converter: V2.0 / 2026-10-26-kda12. Their build numbers are independent. Runtime behavior, audio controls and KK placement described here are supplied by the base GameData; the car-only converter does not upgrade that base.

Kerbal Drift Agency by Roderic Systems. Original KDA synthesized audio and manual text: CC BY 4.0, https://creativecommons.org/licenses/by/4.0/. Retain credit, link the license and indicate changes. Third-party artwork visible in images is excluded. See ASSET-LICENSES.md for exact scope. Runtime code is MIT; converter code is GPL-3.0-only.

Sparky is the Roderic Systems mascot, created with AI assistance. KDA/Roderic Systems/Sparky branding rights are reserved, with permission for unmodified inclusion with KDA and display in screenshots/videos. No endorsement of unrelated products is implied.