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

Start here: give the AI context

You do not need to understand the code. Set up the AI once, then describe what you want in normal language. The separate AI Context guide is for the AI to read.

Step 1 - give it the project

Open the KDA source folder in your coding AI, or attach a ZIP of the repository. For a chat-only AI, attach AI Context.pdf or AI Context.md first; it will also need relevant source files to make actual code changes. A private GitHub link alone may not give it access.

Step 2 - paste this setup prompt

> Help me work on Kerbal Drift Agency (KDA). First read documentation/AI Context.md or AI Context.pdf, then the project README and AGENTS.md if available. Use that context to understand the cars, controller, converter and driving logs. Tell me briefly what files you can actually read. Ask only for anything essential that is missing. Explain things simply and help me collect the right screenshots or driving logs when needed. Do not change anything yet. I will send an easy request from the guide next.

Step 3 - choose a request

Choose New features, Improve something, or Troubleshooting on the following pages. Copy a prompt and replace anything in [brackets]. You can also describe your request in your own words.

Step 4 - let it guide you

Say which car and KSP copy you are using. For handling problems, the AI should help you record a short test. For appearance problems, send a screenshot and the original Automation BeamNG ZIP. If the AI cannot edit files, it should explain the next step.

Starting a new AI conversation? Provide the context document and setup prompt again. A new AI does not automatically remember this project.

1. New features

Use these when you want the mod to do something new. Change the examples to suit your idea.

Long-travel suspension for jumps

> Add adjustable long-travel suspension for [car] so it can handle off-road bumps and land jumps. Keep normal cars working as they do now. Explain the plan simply, implement it, and tell me how to test it.

Working lights

> Add working headlights and brake lights to [car], using its existing lamp fixtures where possible. Let me switch the headlights on and make the brake lights react to braking. Keep the driving behavior the same.

Save my car setup

> Let me save and load named setups for [car], such as Road, Track and Drift. Include the settings that already exist. Explain which settings need the editor and which I can change while driving.

Your own idea

> I want [describe the feature] for [one car / all cars]. The result should feel or look like [describe it]. Use the project context to work out what is needed, explain the plan simply, and help me test it.

You do not need to name code files or invent technical settings. Describe the result you want. The AI should find the right part of the project.

2. Improve something

Describe what feels wrong and what you want instead. The AI should use a useful test before changing behavior.

The car swings from side to side

In [mode], [car] swings from side to side after I release the steering key. Help me record a useful test, find out why, and make the recovery smoother. Keep the other driving modes unchanged.

Make drifting easier to control

I like the drifting in [car], but [I cannot tighten a donut / transitions are too abrupt / the keys do not give enough control]. Help me record this and improve that part without removing the natural slide.

Improve Road or Sport

Make [Road / Sport] feel [calmer / smoother / more responsive] in [car]. Help me compare it at a similar speed before and after. Do not change Race or Drift.

Fix the engine sound

The engine sound in [car] is [too quiet / cutting out / harsh near the limiter]. Help me compare idle, acceleration and high rpm, then improve the sound without changing engine power.

Not sure what to improve?

Here is a driving log from [car]. Explain what the car and controller are doing. Point out the main issue, if any, and suggest one useful improvement. Ask about my intended behavior before making an unclear change.

Use the recording steps on the next page for handling. For sound, include a short video with audio if possible; a driving log cannot show exactly what you heard.

How to record a useful driving test

1. Launch the car in the KSP copy you want to test. Open its driving HUD. If hidden, use Show/hide driving HUD in the body part menu.

2. Tell the AI the car, surface and mode. Mention unusual tire grip, chassis, differential or aid settings. You do not need to type every setting; the log captures many of them.

3. Click RECORD and check for REC and a growing sample count. Drive normally for a few seconds, then reproduce the issue. A focused 20-60 second run is often easier to compare than a long session.

4. Click MARK around the moment you want examined. It adds a timestamp, not video or a written description. If you miss it, give an approximate time and explain what you did.

5. Click STOP RECORDING and allow the writer to finish. Find the newest run inside KDATelemetry, beside KSP_x64.exe in that same KSP installation. Main and test copies have separate log folders.

6. ZIP the whole run folder and attach it, or give a local coding AI access to it. Keep setup.cfg, all CSVs, README.txt and summary.txt together. Explain jumps, crashes, mode changes and deliberate handbrake use.

Paste this with the recording

> This is [car] on [runway / grass / other surface], in [mode]. I was trying to [maneuver]. Near [MARK / approximate time], I pressed [keys] and the car did [what happened]. I wanted it to [expected result]. Check that the log is complete, explain the evidence simply, and guide me through the next useful test or fix.

Keep the same car, surface and settings for a comparison run. Change one thing at a time when practical. The AI should separate jumps and crashes from normal grounded cornering.

3. Troubleshooting: imports and appearance

For an import problem, send the original Automation BeamNG ZIP. Add a screenshot of the issue and, if possible, the look you expected.

A new export will not convert

> Update the converter so it supports this Automation BeamNG ZIP. Here is the export and full error message. Find what is different, keep the correct car data, and explain any remaining limitations simply.

A texture or grille looks wrong

> The [grille / body / window / trim] on [car] should look like [describe it], but it looks like this screenshot. Find the cause and fix this car while keeping its cutouts and small details.

Make a material shinier or more matte

> Make the [paint / wheel / exhaust / interior] on [car] look [shinier / more matte / less metallic]. Keep the color, texture details and transparency. Use the screenshot to identify the right surface.

A suspension component is missing or static

> The [front / rear] [suspension arm / spring / other component] on [car] is missing or does not move correctly. Check the original export and fix its appearance or visual animation. Keep wheel physics unchanged unless a physics change is actually needed.

Want the fix applied automatically to future imports too? Add: "Also update the converter and test a fresh import." Otherwise the AI should keep the request focused on this car.

3. Troubleshooting: the game and next steps

The car does not drive

> [Car] loads but will not drive in [main / test] KSP. Help me check the seat and crew, engine, gear, brakes and missing modules first. Tell me what screenshots or log files you need before changing code.

An error or missing dependency

> KSP or the converter shows this error: [paste the full message]. I am using [KSP copy / converter build]. Help me find the cause and give me simple steps to fix it. Use the README download links for external mods.

Step 5 - test the result

A built update is not automatically installed. Ask where the new files are and which KSP copy they target. When a game update is ready, request installation; the AI should verify KSP is closed, install with a backup and verify hashes without another closure confirmation. If running or unverifiable, leave it ready.

> I tested the change. [Describe what improved and what is still wrong.] Here is the new log or screenshot. Compare it with the previous test and help me with the remaining issue.

If the AI loses context

> Please reread AI Context and the notes from this conversation. Our current goal is [goal]. We already changed [what changed], and the remaining issue is [issue]. Check the current files before continuing.

Keep your original export and before/after logs. Reuse the same AI conversation for related improvements; it should already know how to collect and interpret the evidence.