Batch Rename Toolkit

Documentation

1. Introduction

Batch Rename Toolkit is an Unreal Editor plugin that helps teams enforce consistent Content Browser asset naming. It scans folders under /Game, checks each asset against built-in Unreal-style prefix rules (such as T_ for textures, M_ for materials, and BP_ for Blueprints), and highlights assets that do not follow your studio conventions.

The tool is designed for technical artists, environment artists, and engineers who need to clean up large content libraries before milestone delivery, Fab submission, or handoff to another team. Instead of manually renaming assets one by one, you can validate an entire folder tree, preview changes, apply bulk renames safely, and export reports for review.

Key capabilities include: folder-based scanning with optional subfolder recursion; a detailed results table showing asset name, type, path, expected prefix, and suggested fix; preview before rename with conflict detection; bulk rename of all violations or only selected rows; single-file rename with custom names; session-based undo; Content Browser sync; clipboard and CSV export; automatic redirector fixup and Blueprint recompilation after renames.

Requirements: Unreal Engine project with the plugin installed and enabled. The plugin is Editor-only and does not affect packaged game builds. Supported asset types are listed in Section 3.7; assets without a configured rule appear as "No Rule" in the results table.

Created by 300Mind. Version 1.0.

2. How to Install

2.1 Installation Steps

Step 1 - Copy the plugin: Place the entire BatchRenameToolkit folder inside your project Plugins directory. Example path: YourProject/Plugins/BatchRenameToolkit/. The folder must contain BatchRenameToolkit.uplugin, Source/, and Binaries/ (if pre-built).

Step 2 - Generate project files (if needed): Right-click YourProject.uproject and choose "Generate Visual Studio project files" when building from source on a new machine.

Step 3 - Enable the plugin: Launch Unreal Editor, open Edit > Plugins, search for "Batch Rename Toolkit", check Enabled, and accept the restart prompt.

Step 4 - Verify the menu: After restart, confirm Tools > Batch Rename Toolkit appears in the main menu. Selecting it opens the validator panel as a dockable editor tab.

Step 5 - First scan: Open the tool, leave the default folder as /Game or enter a subfolder such as /Game/Environment, then click Scan to confirm assets are detected.

2.2 Building from Source

If Binaries are not included, compile the plugin with your project in Visual Studio, or use RunUAT BuildPlugin for Fab-style packaging validation. The plugin module is BatchRenameToolkit (Type: Editor).

Ensure Visual Studio 2022 with the "Game development with C++" workload is installed. Match the MSVC toolset to your Unreal Engine version.

2.3 Troubleshooting Installation

Plugin not visible in Edit > Plugins: Confirm the folder name matches the module name and that BatchRenameToolkit.uplugin is at the plugin root, not nested incorrectly.

Compile errors: Open the Output Log and rebuild the project. Verify you are using a compatible Unreal Engine version for this plugin build.

Menu entry missing: Check that the plugin is enabled for your project (not only installed globally) and restart the editor after enabling.

3. How to Use

3.1 Opening the Tool

Open Tools > Batch Rename Toolkit from the main menu. The panel title displays "Unreal Asset Naming Prefix Validator". You can dock the tab alongside the Content Browser for a convenient workflow. The panel is divided into: controls (top), results table (left), naming rules reference (right), and action buttons (bottom).

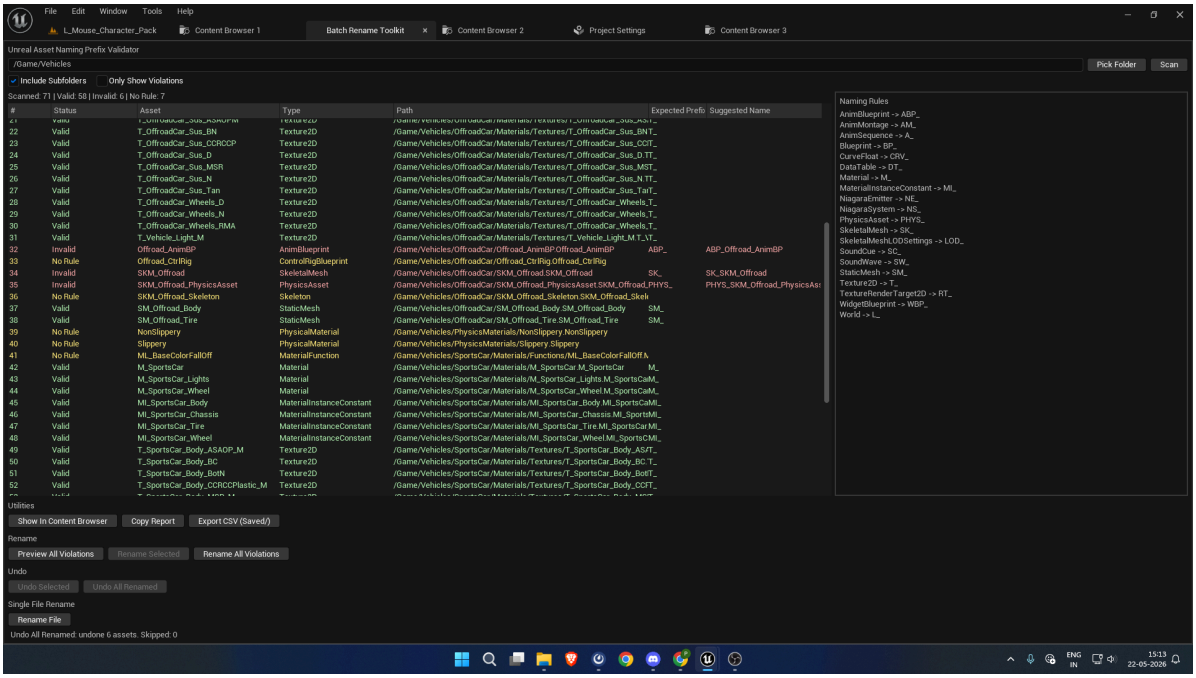

Figure 1: Main validator panel showing scan summary, results table, naming rules, and action buttons.

3.2 Scanning Assets

Folder path: Enter a valid long package path (e.g. /Game/Props) or click Pick Folder to choose a directory in the Content Browser tree. Paths like Content/MyFolder are normalized to /Game/MyFolder automatically.

Include Subfolders: When checked (default), the scan uses recursive package paths and finds assets in all child folders. Uncheck to scan only the exact folder.

Scan button: Queries the Asset Registry for all assets under the selected path. A summary line shows: Scanned count, Valid, Invalid, and No Rule totals.

Only Show Violations: Filters the table to hide assets that already pass validation (have a rule and correct prefix). Useful for large libraries when you only need to fix problems.

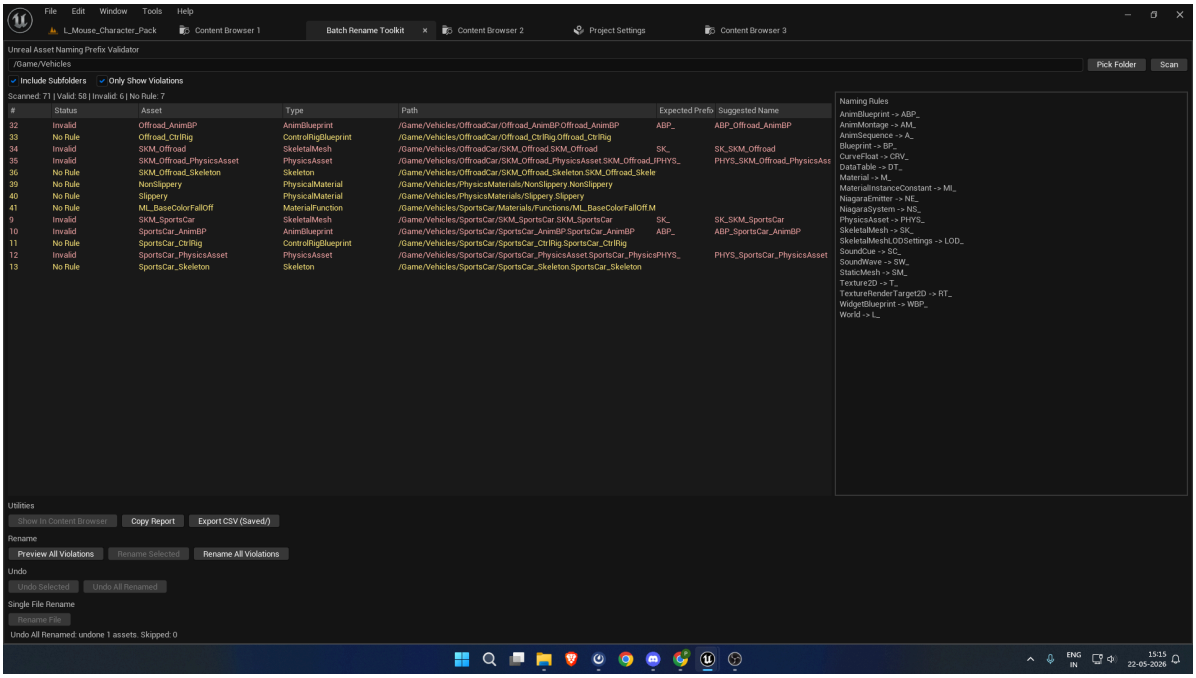

Figure 2: Only Show Violations enabled — table filtered to Invalid and No Rule assets only.

3.3 Understanding the Results Table

Each row represents one asset. Columns: # (display index), Status, Asset (name), Type (class), Path (object path), Expected Prefix, Suggested Name.

Status values: Valid (green) — name starts with the correct prefix. Invalid (red) — rule exists but prefix is wrong; Suggested Name shows the proposed fix. No Rule (yellow) — asset class is not in the built-in prefix list.

Color coding helps you prioritize Invalid rows for renaming. Select one or more rows with Ctrl/Shift click for targeted operations.

The Naming Rules panel on the right lists all built-in class-to-prefix mappings for quick reference during review.

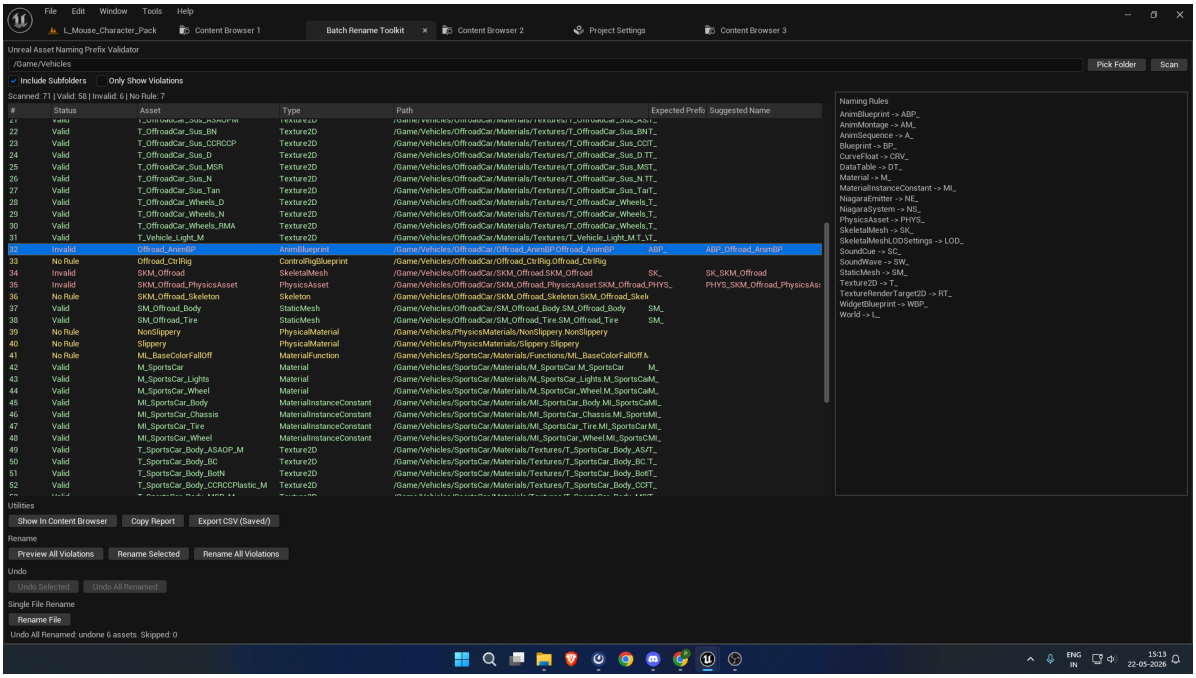

Figure 3: Selecting an Invalid asset row (e.g. AnimBlueprint Offroad_AnimBP) to inspect Expected Prefix and Suggested Name.

3.4 Renaming Workflows

Preview before rename

Preview All Violations: Builds a rename plan for every Invalid asset without applying changes. Opens a log window listing planned OldPath -> NewPath entries, plus any conflicts or skips (duplicate targets, existing assets, unload failures). Always preview before large bulk operations.

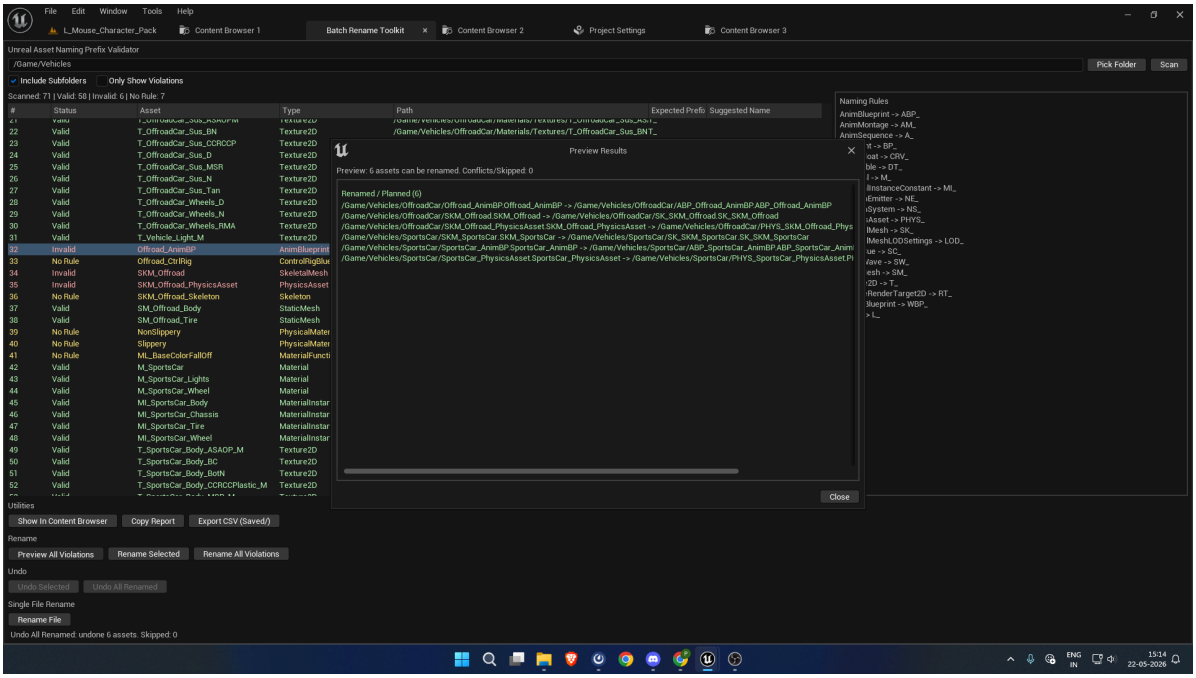

Figure 4: Preview All Violations dialog listing planned renames before applying changes.

Rename selected or all violations

Rename Selected: Renames only the rows you have highlighted, provided they are Invalid with a non-empty suggested name. Useful for partial fixes.

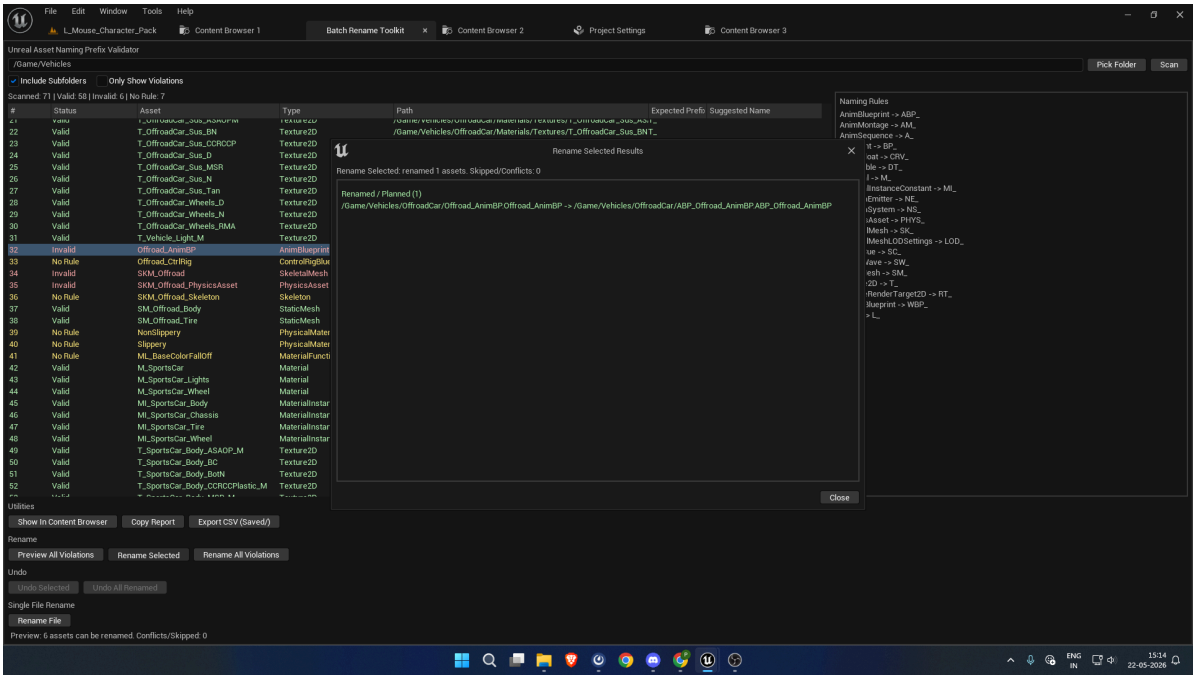

Figure 5: Rename Selected results confirming a successful rename (Offroad_AnimBP → ABP_Offroad_AnimBP).

Rename All Violations: Applies renames to every Invalid asset in the current scan. Shows a progress dialog, then a results popup with success, warning, and error sections.

Single-file rename

Rename File: Available when exactly one row is selected. Opens a dialog to type a custom name, with an optional checkbox "Keep/Add default prefix if missing" to prepend the expected prefix when needed. Shows a live Final Name Preview before confirming.

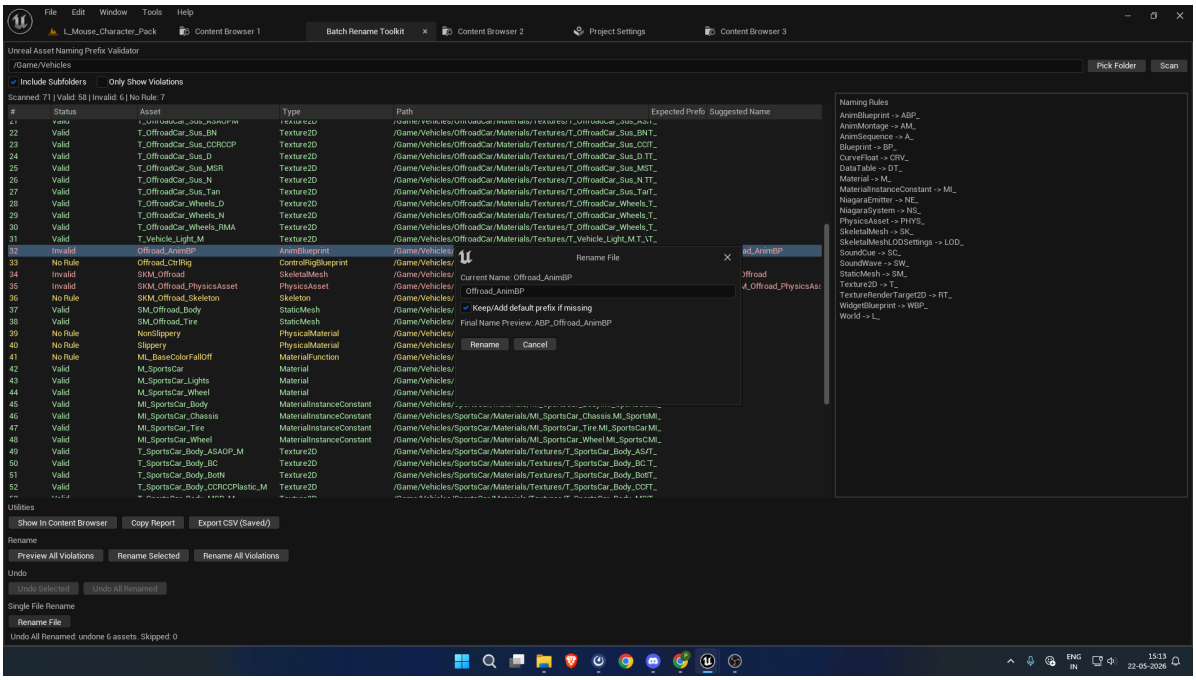

Figure 6: Rename File dialog with custom name and Keep/Add default prefix option.

After any rename: The tool fixes redirectors in the scanned folder, recompiles affected Blueprints, refreshes the scan, and records undo data for the session.

How suggestions work: For Invalid assets, the suggested name is ExpectedPrefix + CurrentAssetName (prefix is prepended to the full existing name). Example: texture "WoodAlbedo" becomes "T_WoodAlbedo". The tool does not strip partial or incorrect prefixes before prepending.

3.5 Utilities

Show In Content Browser: Syncs the editor Content Browser to the currently selected asset(s) in the table.

Copy Report: Copies the filtered table as CSV-formatted text to the clipboard (Status, AssetName, Class, Path, ExpectedPrefix, SuggestedName).

Export CSV: Writes the same report to YourProject/Saved/BatchRenameToolkitReport.csv. Share this file with leads or use it in spreadsheet reviews.

3.6 Undo

Undo Selected: Reverts selected assets to their original names, but only if they were renamed by this tool during the current editor session.

Undo All Renamed: Reverts every asset tracked in the session undo list.

Important: Undo history is stored in memory only. Closing Unreal Editor clears undo records. Undo is blocked if the original name is now occupied by another asset.

3.7 Supported Prefix Rules (Built-in)

Texture2D = T_ | Material = M_ | MaterialInstanceConstant = MI_ | StaticMesh = SM_ | SkeletalMesh = SK_ | Blueprint = BP_ | AnimBlueprint = ABP_ | AnimSequence = A_ | AnimMontage = AM_

SoundWave = SW_ | SoundCue = SC_ | NiagaraSystem = NS_ | NiagaraEmitter = NE_ | World = L_ | WidgetBlueprint = WBP_ | PhysicsAsset = PHYS_ | SkeletalMeshLODSettings = LOD_

TextureRenderTarget2D = RT_ | CurveFloat = CRV_ | DataTable = DT_

Prefixes are normalized to include a trailing underscore. Custom rules are not editable in the UI in v1.0; contact support for studio-specific rule sets.

3.8 Known Limitations

Editor-only: No runtime or packaged-build functionality.

Hardcoded rules: Prefix map is defined in plugin source; the UI displays rules but does not allow editing them.

Prepend-only suggestions: Wrong prefixes are not removed before adding the correct one; review Preview output carefully for edge cases.

Session undo only: No persistent undo across editor restarts.

No Rule assets: Classes not in the table are not auto-renamed; extend rules in a future version or via custom support.

Large batches: Very large folders may take time to scan and rename; use "Only Show Violations" and rename in smaller selections if needed.

4. Support

For bug reports, feature requests, or custom prefix rule sets for your studio pipeline, contact 300Mind via the support link on the Fab product page or visit https://300mind.studio.

Website: https://300mind.studio | Discord Server ：https://discord.gg/wUSAwrWyKC
We appreciate your feedback and Fab reviews.

Thank you for choosing Batch Rename Toolkit. We hope this documentation helps you maintain a clean, professional Content Browser. Happy developing!

THANK YOU