XRVP COLLECTIVE · USER GUIDE

BAIBv2+MCP

Blueprint AI Bridge, Multiplied

Blueprint AI Bridge upgraded with RT2 round-trip compression, a 1052-tool MCP server across 22 domains, a resilient gateway, and full editor automation for AI coding agents.

Compress Blueprint graphs for AI, or let an MCP-enabled agent create and edit Blueprints, Widgets, Niagara VFX, materials, levels, physics, audio, landscapes, foliage, and PCG graphs directly.

UE 5.2 – 5.8 · Windows · Editor only.

ENGINE	PLATFORM	TOOL COUNT	CONTACT
UE 5.2 - 5.8	Windows (editor only)	1052 tools · 22 domains	fab.xrvp@gmail.com

Contents

1	Overview	3
2	Installation & Setup	4
3	Toolbar Workflow — Blueprint to AI	5
4	Panel Modes & RT2 Format	6
5	MCP Server & Gateway	8
6	Tool Domains (22)	9
7	Compact MCP Facade & Configuration	10
8	Security, Safety & Parallel Agents	11
9	Practical Workflow Guidance	12
10	Known Limitations	13
11	Feedback & Roadmap	14

INTRODUCTION

Overview

BAIBv2+MCP is an Unreal Editor plugin for exporting Blueprint graphs into compact, AI-editable text (**AIBP-RT2**), applying AI-edited results back into Unreal, and exposing local editor tools to CLI agents and MCP clients. It builds on the original BAIB compression workflow and adds a full agentic tool surface: **1052 MCP tools** across **22 domains**, a standalone gateway that survives editor restarts, and hardened security defaults.

Two ways to work: hand-copy compressed Blueprint text to any chat AI, or point an MCP-enabled agent (Claude Code, Gemini CLI, Cursor) at the gateway and let it create, wire, and compile Blueprints, Widgets, VFX, materials, and levels directly.

WHAT IT IS

An editor plugin that gives AI agents a structured, compressed view of Blueprint graphs plus 1052 typed tools for the rest of the editor — assets, widgets, VFX, levels, physics, audio, and more.

WHY IT EXISTS

Raw Blueprint clipboard data and ad-hoc Python are too noisy and unsafe for AI-driven editing. BAIBv2+MCP gives agents typed, validated, undo-able operations instead.

Core upgrade over BAIB v1

- ✓ AIBP-RT2 with explicit `WIRES` manifests for reliable round-trip reconstruction
- ✓ Local HTTP tool server at `127.0.0.1:3777` inside the live editor process
- ✓ Standalone Python gateway at `127.0.0.1:3778` that survives editor restarts and crashes
- ✓ Included stdio MCP bridge for external clients (Claude Code, Cursor, Gemini CLI)
- ✓ 1052 typed tools across 22 domains — Blueprint, Widget, Niagara, Material, Level, Sequencer, Animation, Rendering, Gameplay, Physics, Audio, Landscape, Foliage, PCG, and more

GETTING STARTED

Installation & Setup

1. **Install the plugin.** Copy the `BAIBv2_MCP` folder into your project's `Plugins/` directory.

2. **Enable it.** Open the project, go to Edit → Plugins, search "BAIB", confirm it's enabled, and restart the editor.

3. **Start the gateway.** Run `Tools/BAIBGateway/Start-BAIBGateway.bat` from the plugin directory. This Python process handles tool routing and stays alive even when Unreal restarts or crashes.

4. **Connect your agent.** Point your AI agent (Claude Code, Gemini CLI, Cursor, or anything MCP/HTTP-capable) at the gateway URL: `http://127.0.0.1:3778`.

CRASH WATCHDOG RUNS AUTOMATICALLY

The plugin auto-launches `crash_watchdog.py` every time the editor starts, monitoring for crashes. Recovery is faster with the gateway's own `/lifecycle/ensure_running` and `/lifecycle/restart` endpoints, which relaunch the editor with `-AutoDeclinePackageRecovery` and clear the stale crash-restore prompt — no manual clicking required.

Check `Tools/BAIBGateway/logs/crash_watchdog.log` for activity. Opt out with `r.BAIB.WatchdogEnabled=0` in `DefaultEngine.ini [ConsoleVariables]`.

First tool call

Ask your agent to "get context from Unreal." It should call `get_context` and return a summary of your open assets and selected actors — a good smoke test that the gateway, editor, and bridge are all wired together correctly.

Requirements

SETTING	
Engine	UE 5.2 -5.8
Platform	Windows editor build (editor only, no runtime cost)
Python	PythonScriptPlugin enabled for `run_python` / `run_python_safe`

STEP-BY-STEP

Toolbar Workflow — Blueprint to AI

Open a Blueprint editor. BAIBv2+MCP adds toolbar actions and matching menu entries under **Tools → BAIB — Blueprint AI Bridge**. This is the manual path — useful even without an MCP-connected agent.

ACTION	WHAT IT DOES	SHORTCUT
Copy for AI	Exports selected Blueprint nodes as AIBP-RT2, updates the BAIB panel, and copies the payload to the OS clipboard.	`Ctrl+Shift+C`
Paste from AI	Reads returned RT2 from the clipboard, reconstructs Unreal Blueprint clipboard text, and pastes it into the focused graph.	`Ctrl+Shift+V`
Panel	Opens the dockable BAIB panel for inspection, conversion, warnings, prompts, and alternate output modes.	`Ctrl+Shift+B`

Blueprint to AI

1. Open a Blueprint graph and select the nodes you want to edit or explain.
2. Click **Copy for AI** or press `Ctrl+Shift+C`.
3. Paste the generated AIBP-RT2 into your AI, along with your request.
4. Ask the AI to return valid AIBP-RT2 only, with no Markdown fencing.

AI back to Blueprint

1. Copy the AI's returned AIBP-RT2 text.
2. Focus the target Blueprint graph in Unreal.
3. Click **Paste from AI** or press `Ctrl+Shift+V`, then run `refresh_blueprint_nodes` and compile.

For event-driven graphs, BAIB clears Unreal's stock disabled-placeholder state from imported event nodes, so a connected generated `Event Tick` or `BeginPlay` does not silently remain disabled.

ENCODING

Panel Modes & RT2 Format

MODE	USE
RT2	Full round-trip format for AI edits and paste-back reconstruction.
Advice	Readable Blueprint context for questions, explanation, and review.
Advice+	Advice output plus local function bodies where available.
ASCII	Plain-text execution flow diagram.
Box	Box-drawing graph summary.
Pseudo	Plain-language pseudocode.
Mermaid	Mermaid flowchart output for docs and knowledge bases.

RT2 format

RT2 adds a `WIRES` manifest before the node blocks, making graph connections explicit, easier to validate, and far less likely to reconstruct with ghost or missing wires.

```
AIBP-RT2
GRAPH "EventGraph"

WIRES
N1.then -> N2.execute
END WIRES

NODE N1 TYPE=K2Node_Event EVENT=ReceiveBeginPlay
PIN OUT then exec

NODE N2 TYPE=K2Node_CallFunction FUNC=Script/Engine.KismetSystemLibrary.PrintString
PIN IN execute exec
PIN IN InString string = "Hello World"
PIN OUT then exec
```

Rules for AI edits

- ✓ Keep existing `N#` ids unless intentionally replacing nodes.
- ✓ New nodes use the next unused id.
- ✓ Keep the `WIRES` section consistent with the pin connections.
- ✓ Return plain AIBP-RT2 only — no Markdown fences.

RT2 IS CREATION-ONLY

`apply_rt2` is for new, empty graphs only. Editing an existing graph must use surgical tools — `place_node`, `connect_nodes`, `delete_node`, `set_node_pin_default` — since a full graph replacement loses UE5 internal state (GUIDs, struct bindings, component variable references).

AUTOMATION

MCP Server & Gateway

When Unreal Editor is running and BAIB loads, the plugin starts a local HTTP tool server inside the live editor process. A separate, standalone Python gateway sits in front of it so agent connections survive editor restarts.

EDITOR SERVER — PORT 3777

Lives inside the Unreal Editor process. Executes every tool against the live world and asset registry. Goes down when the editor restarts or crashes.

GATEWAY — PORT 3778

Standalone Python process. Stays alive across editor restarts, supervises the editor lifecycle, and serves doc-index lookups even when the editor is offline.

Endpoints

ENDPOINT	PURPOSE
`GET /mcp/health`	Health check — `{"status":"ok","server":"BAIB MCP"}`
`GET /mcp/tools/list`	Full tool schema listing.
`POST /mcp/tools/call`	Invoke a tool by name with arguments.
`GET /lifecycle/status`	Gateway-only — editor process health.
`POST /lifecycle/ensure_running`	Gateway-only — launch or confirm the editor is up.
`POST /lifecycle/restart`	Gateway-only — restart the editor cleanly, suppressing the crash-restore modal.

MCP stdio bridge

The plugin includes a stdio MCP bridge at `Tools/MCP/baib_mcp.py`. MCP clients launch that script and it forwards tool calls to the gateway.

```
{
  "mcpServers": {
    "baib": {
      "command": "python",
      "args": [
        "E:/.../Plugins/BAIBv2_MCP/Tools/MCP/baib_mcp.py",
        "--baib-url",
        "http://127.0.0.1:3778"
      ]
    }
  }
}
```

REFERENCE

Tool Domains (22)

BAIBv2+MCP organizes its 1052 tools into 22 specialized domains, each backed by a dedicated dispatcher module for fast schema caching (1,148× faster than a monolithic dispatch table) and safe parallel-agent workflows.

DOMAIN	DESCRIPTION
Blueprint	Graph I/O (RT2), logic wiring, refactoring, static analysis.
Widget	UMG hierarchy management, slot layout, theming, animation.
Material	Graph construction, instance parameter sweeps, thumbnail rendering.
Niagara	System/Emitter authoring, module stack management, VFX archetypes.
Level	Actor spawning, transforms, World Partition, lighting/volumes.
Sequencer	Level sequence creation, track/keyframe management, CineCamera rigs.
Animation	AnimBP/Montage creation, skeleton inspection, rig tracks.
Rendering	Post-process volumes, Lumen/Nanite settings, GPU profiling.
Gameplay	Enhanced Input, AI (Behavior Trees/Blackboards), GAS, Chaos.
Reflection	Native UClass/UFunction inspection, C++ skeleton generation.
Production	Build management, headless testing, Horde/Zen/Multi-User status.
Search	Unified find, reference tracking, pin provenance.
Asset	Core CRUD (duplicate, rename, move, delete), redirector management.
Physics	Collision profiles, constraints, body properties, simulation state.
Audio	Sound cues, attenuation, audio components, EQ, reverb.
Landscape	Creation, layer painting, material assignment, sculpting.
Foliage	Foliage type creation, instance painting, density, cull distance.
PCG	Procedural Content Generation graph authoring and sampling recipes.

Physics, Audio (MetaSounds graph editing), and 4 Landscape spline tools have a small number of engine-limited stubs on UE 5.7 — see Known Limitations.

Tier-3 intent tools

High-level "intent" tools use an in-editor planner to execute complex multi-step tasks from a single prompt: `make_vfx`, `add_logic`, `style_widget`, `spawn_archetype`, plus `discover_intent` for finding the right one.

CONFIGURATION

Compact MCP Facade & Configuration

1052

tool schemas, curated down to a small default surface

The stdio bridge exposes a compact core tool set plus three meta-tools — `search_tools`, `get_tool_schema`, `call_tool` — by default, so 1052 tool schemas never blow up your client's context window. Set `BAIB_MCP_FULL_TOOLS=1` to expose everything.

Editor MCP server (port 3777)

VARIABLE	EFFECT
`BAIB_AUTH_TOKEN`	Pin the security token for CI/automation. Otherwise a per-session token is generated and written to `Saved/BAIB/session_token.txt`.
`BAIB_ALLOW_ORIGIN`	Exactly one browser origin allowed to call the server. Default: all browser-origin requests get 403.
`BAIB_MAX_RESPONSE_BYTES`	Response size cap (default 5 MB). Larger payloads are saved to disk and replaced with a small JSON error.
`BAIB_PYTHON_ENABLED=1`	Required for the raw `run_python` tool.
`BAIB_PLANNER_ENABLED=1`	Enables Tier-3 intent routing (default: on).
`BAIB_RC_PORT`	Override the Remote Control fallback port (default: 30010).

MCP stdio bridge

VARIABLE	EFFECT
`BAIB_MCP_FULL_TOOLS=1`	Expose all 1052 tool schemas instead of the compact core set + meta-tools (default: compact).
`BAIB_TOKEN_FILE`	Override the path the bridge reads the session token from.

Gateway editor lifecycle

VARIABLE	EFFECT
`BAIB_UE_ROOT` / `BAIB_PROJECT_FILE`	Engine install and `.uproject` for supervised launches (auto-detected when unset).
`BAIB_AUTO_RELAUNCH=1`	Auto-relaunch the editor after a crash (default: off).
`BAIB_SUPPRESS_CRASH_RESTORE`	Suppress the "restore crashed session?" modal on relaunch (default: on).
`BAIB_LIFECYCLE_POLL_INTERVAL` etc.	Health-poll tuning (defaults: 30s poll / 3 fails / 600s launch timeout / 45s zombie grace).

SAFETY

Security, Safety & Parallel Agents

SECURE BY DEFAULT

Every tool call is gated by a per-session `X-BAIB-Token`. Browser-origin requests are rejected with 403 (no CORS headers by default). The token is written to `Saved/BAIB/session_token.txt` for local bridges to read.

RESPONSE SIZE GUARD

Responses over 5 MB are parked to disk (`Saved/BAIB/oversize_<tool>_<timestamp>.json`) instead of wedging the calling client, and replaced with a small JSON pointer error.

Transactions & confirmation

- ✓ For complex multi-step changes, agents use `begin_transaction` / `commit_transaction` so the entire operation can be Undone in one go.
- ✓ Deleting assets or actors requires an explicit `confirm: true` flag to prevent accidental data loss.
- ✓ BAIB automatically validates most changes; on failure, agents call `get_compile_errors` / `explain_compile_error` to self-heal.

Parallel agent workflows

Every mutating tool call is serialized through a mutation FIFO queue, so multiple agents (or one agent firing bursts of calls) can safely write to the world concurrently without corrupting editor state. Read-only tools — including the off-screen perception tools below — bypass the queue entirely since they never touch live state.

Off-screen perception

`capture_viewport_offscreen` and `verify_visual` render through a transient `SceneCapture2D` instead of reading the OS back-buffer, so they work while the editor is minimized, unfocused, or fully backgrounded — unlike the older `screenshot_capture`/`capture_viewport` tools. `verify_visual` additionally reports per-actor frustum visibility and normalized screen-space bounding boxes, so an agent can check placement work programmatically instead of eyeballing a screenshot.

PRACTICAL GUIDANCE

Practical Workflow Guidance

- ✓ Use RT2 for EventGraph behavior; use surgical tools (`place_node`, `connect_nodes`) for edits to existing graphs — RT2 is creation-only.
- ✓ For complex actors, prefer a reusable native parent class and create a Blueprint child through the tool surface.
- ✓ Call `get_blueprint_summary` first at the start of a Blueprint-editing session — it's a one-shot consolidated JSON snapshot that replaces 5-6 separate inspection calls.
- ✓ Always run `refresh_blueprint_nodes` after `create_blueprint`, `place_node`, `connect_nodes`, `delete_node`, or `apply_rt2`, before the final compile — Unreal can keep stale pin metadata on freshly created or reconstructed nodes.
- ✓ For Widget Blueprints, the Designer tree is not RT2 — use the dedicated widget tools (`add_widget`, `set_widget_property`, `bind_widget_event`, `apply_widget_spec`).
- ✓ Treat `run_python` / `run_python_safe` as a local power tool — it executes code inside the live Unreal Editor process.

Example prompts

DOMAIN	EXAMPLE
Blueprint	"In /Game/Blueprints/BP_Player, add a Print String node after the Tick event that prints 'Hello from AI'."
UI	"Create WBP_MainMenu with a button centered on screen that opens level 'Map_Start' when clicked."
VFX	"Create a sparking Niagara system at /Game/VFX/NS_Sparks."
Level	"Spawn 10 cubes in a circle around the player."
Materials	"Create a glowing red material and apply it to all selected actors."

LIMITATIONS

Known Limitations

ENGINE-LIMITED STUBS (UE 5.7)

A small number of tools cannot be implemented against public UE 5.7 APIs and return `NOT_SUPPORTED`:

• Physics: `apply_voronoi_fracture` — FractureEditor is UI-only.

• Audio: 15 MetaSounds graph-editing tools — Builder API uncertain in UE 5.7 (19 other Audio tools are fully implemented).

• Landscape: 4 spline tools — protected-member access blocks a public implementation (23 other Landscape tools are fully implemented).

TechDebt.md tracks each stub with its resolution path; check it before assuming a gap is permanent.

Other limitations

- ✓ PCG tools require the PCG plugin to be enabled in the project.
- ✓ Unknown Blueprint node families are best-effort and may warn during scan or reconstruction.
- ✓ Custom Blueprint-generated widget classes (WBP_*) cannot be nested via `add_widget` — it only resolves native C++ UWidget subclasses; build with native placeholders, then drag-drop WBP instances in the Designer.
- ✓ This is an editor-focused workflow plugin, not a runtime gameplay plugin — no runtime cost.
- ✓ Direct HTTP use works well for CLI agents; MCP clients should use the included stdio bridge.

SUPPORT

Feedback & Roadmap

BAIBv2+MCP is under active development — new tool waves land regularly across all 22 domains. Send outputs, broken reconstructions, or anything surprising; the more examples received, the faster improvements ship.

CONTACT

fab.xrvp@gmail.com — send your compressed output, the AI response, or a description of the tool call that misbehaved to help shape the next release.

Generated for XRVP Collective from the BAIBv2_MCP plugin docs and implementation notes.