PadForge 5.2 reviewer field guide

Practical workflows with observable results and stated limits

Use this guide

PadForge 5 combines physical input sources with selectable Windows outputs. A useful review follows the complete path: an input the device supplies, a mapping or action PadForge evaluates, an output the recipient accepts, and any feedback that returns to the physical device. The same controller can participate in several paths, and their limits differ.

Start with workflow 1. Then choose the workflows for equipment and software already available. Each has a defined observation rather than an assumed success. Mark an unavailable route skipped, a configuration-only check configured, and a completed receiver or hardware check observed. The procedures below follow the PadForge documentation and public sources. They were not run while this guide was prepared.

This guide describes PadForge 5.2.0 and its documentation as of October 27, 2026. Record the installed build before you start. The links go to the live documentation, which can change after this guide. A different build or dependency needs its own check against the documentation.

The PadForge film and its landscape, portrait and square downloads are at padforge.org/press/#film.

Review question	Start here	Required observation
Can it replace an ordinary remapper for one game?	1, 2, 6, 7 and 8	Recipient input, release behavior, profile selection and any translation gaps
What do motion and unusual inputs add?	3, 4, 5, 12, 13, 16 and 17	Named source values and the chosen output consumer
How far does feedback travel?	9, 10, 11 and 15	Physical effect or audio, with the exact game and transport
Can a phone or another PC join the setup?	14 and 15	Input at the destination and release on loss of the source
What still depends on vendor software or a specialist tool?	17 and 18	The documented dependency and a distinct receiving result
What does 5.1.0 add for mice, keyboards and accessibility?	18, 19 and 20	The lit or vibrating device, the typed or pointed output, and what a screen reader speaks

Review question	Start here	Required observation
What does 5.2.0 add?	21, 22, 23, 24 and 25	A telemetry reading or mod effect at the controller, haptic rumble on a named pad, a DualShock 4 route, and Falcon force or a screen picture

Sources: Type buttons / Auto-mapping / Beyond gamepads / The PadForge film

Prepare a reversible review setup

Create and load a separate named profile before editing. Fully exit PadForge through its tray menu before taking a cold copy of PadForge.xml and its referenced assets. Closing the window can leave the engine running. Keep the original application configuration, recipient bindings, and hiding choices for rollback.

Use the current driver and prerequisite instructions for the exact output. HIDMaestro installs on demand when an enabled Xbox, PlayStation, Nintendo or Extended slot has an online assigned source and the engine runs. An empty slot alone is insufficient. The HIDMaestro card lights its flame while the driver is in Windows' driver store and shows the version PadForge carries, which is not proof that a virtual controller is live. It reads Not Installed until the first virtual controller puts the driver there. A failed setup leaves "Error: Failed to initialize HIDMaestro." in the status bar and runs again when the slot is turned off and on, its type or profile changes, or its controller reconnects. A Keyboard + Mouse half set to a keyboard or mouse model also installs HIDMaestro on first use.

Composite Sony, Nintendo and Valve presets also install usbip-win2's virtual USB host controller on first use, with no prompt and no USB hub restart. Record that dependency separately. Do not replace a working transport with a random newer upstream build as part of a comparison.

Record which MIDI path the PC runs. The Windows MIDI Services card in Settings names the API built into Windows, the runtime PadForge installs on Windows 11 24H2 or later, or the legacy MIDI API. Under the legacy API a MIDI slot creates no port, so it needs an existing Output Port such as a loopback driver.

- Record build, Windows version, architecture, physical model, firmware where relevant, cable or radio path, output preset, recipient version, and any Steam Input or vendor-software setting.
- Keep one intended output visible to the recipient while establishing the first mapping. Use the documented HidHide setup only if the physical and virtual devices otherwise produce double input. Verify PadForge still receives the physical source.
- Keep keyboard, mouse, voice, and tag-triggered actions in a disposable text document or controlled recipient. A macro that sends keys follows focus.
- The sixteen-slot total is shared across seven families. XInput has four controller positions. DSU serves slots 1 through 4. A single VR slot supplies both virtual hands.
- To leave no driver behind on a disposable review PC, use Uninstall on the HIDMaestro Driver and BthPS3 Bluetooth Driver cards in Settings after the review. Each lists what it removes and keeps before it runs. On the HIDMaestro card, the option Also remove usbip-win2 set up by its own installer can take off usbip-win2's root-hub filter, and that restarts every USB 3 root hub. Do not select it while other USB devices are in use.

- After an update from 5.1.0, the first virtual controller that needs it installs the HIDMaestro 2.1.0 driver once.

Sources: Status indicators / Microsoft MIDI releases / Language, appearance and window / Microsoft MIDI artifact removal / HIDMaestro / Extended Controllers Not Working / First-run auto-install

Session A One driving profile with a held command menu

Review one practical setup from start to finish: ordinary driving controls stay available, a held layer opens game commands, foreground selection loads the right profile, and feedback is checked separately.

Equipment and preconditions

- Required: one gamepad, an owned driving game that accepts its chosen output, and a separate review profile. Optional: pedals, a force-feedback wheel, or a shaker.
- Use a running engine, enabled slots and online assigned sources. Check the first-install prerequisites. Record the game build, output preset and current bindings. Use a practice session.

Procedure

1. Create the game output and verify steering, throttle and brake in the game. If adding pedals, assign them to the same slot and check every source separately before combining them.
2. Create a separate Keyboard + Mouse slot for the same gamepad. Add a Hold layer on left bumper and a named-layer radial menu. Use Layer Holds the Menu Open and Right Stick under Steer With. Reserve an activator that the game does not use: consumption on this slot does not automatically mute its mapping on the other output.
3. Give one cell the actual keyboard binding the game uses for a harmless menu action. Give another an explicitly Any Layer macro with one deliberate key action. Use On Click while the layer remains held.
4. Save the named setup, add the actual game executable, and enable foreground switching. Check the active profile when entering and leaving the game. Load Default manually first if an external hold is active.
5. Drive, hold the command layer, select and click a cell, release, and resume driving. Record the game action and clean return of normal controls. If the game rejects mixed keyboard/controller input, record that boundary. While the menu is open, also check that its steering and click do not trigger unintended actions on the other gamepad output. Use an input the game does not need during command selection, or record the conflict as a partial result.
6. If feedback equipment is available, trigger one repeatable game event. Record controller rumble or wheel force separately, then add shaker routing only after that game-feedback path is established.

Result and evidence

A useful session establishes the complete profile, input, menu, macro and receiving-game path. The optional feedback result is a separate observation. Neither a highlighted cell nor a profile match proves that its command reached the game.

- Keep the source assignment, two slot identities, active profile, Hold/menu/macro settings, actual game binding, receiving action and release outcome.
- Record optional pedal and feedback branches only when performed.

Limits and reset

- This uses On Click with an eligible Any Layer macro. Since 5.1.0 a menu set to On Touch Release also runs its hovered cell when its layer ends, which needs its own check.
- A profile change can recreate output when identity, enabled state or topology changes. The vendor rows that replaced the 5.0.0 mirror and Sensa switches are device assignments and follow the profile.
- Fully exit through the tray before returning to the old remapper, restore game bindings and hiding choices, and load the preserved profile.

Sources: Auto-mapping / Multi-slot device assignment / Activator dialog and modes / Style, host, fire mode and cells / Contents and auto-switch / The old Dashboard switches / Per-layer activators

Session B Shared game input with a local phone command deck

Use a controller on one PC to play on another, while a phone provides commands directly to the receiving PC. This connects two supported routes without assuming Web Menus crosses Remote Link.

Equipment and preconditions

- Required: two PadForge PCs on a controlled network, a gamepad on the owner, a phone/browser, and an accepting game or input recipient on the consumer. Optional: a physical feedback-capable pad and a game rumble event.
- Use running engines, enabled destination slots and online assigned input. Record both builds, the pairing restriction, browser version and recipient.

Procedure

1. Pair the two PCs and assign the owner's shared gamepad to the consumer's game output. Establish ordinary input and release in the recipient.
2. On the consumer, create a Touch Grid menu on a local command slot with a harmless action and enable Show on Web Controller. Start that PC's Web Controller listener.
3. Connect the phone directly to the consumer's displayed address, choose Web Menus, and assign the phone to that command slot. Keep its menu and macro scopes eligible.
4. Use the shared gamepad for ordinary input, then tap a phone command. Check the action on the consumer, not on the controller-owner PC. If using foreground profiles, confirm that the active profile contains both intended slots.
5. Trigger optional game rumble on the consumer and observe the owner's physical pad. Then interrupt the shared-input path and phone path one at a time and record their separate release/reconnect outcomes.

Result and evidence

A passing session has shared controller input at the consumer, a phone command executed on that same host, and an optional physical feedback result at the owner. Its two loss/reconnect observations remain distinct.

- Record PC roles, virtual outputs, source assignments, phone host, layer/profile state and actual recipient actions.
- Keep pairing codes, access codes, QR contents and private addresses out of the shared result.

Limits and reset

- Web Menus is hosted directly on the consumer here because Remote Link does not share it.
- The gamepad-only restriction is not a general sandbox. Browser permission and feature support remain platform-specific. No game compatibility is invented by this composition.
- Restore assignments, stop trial listeners, revoke temporary permissions or pairing as appropriate, and exit fully through the tray before reverting the earlier setup.

Sources: Setup and network / Input and feedback / Web Menus (#471) / Motion permission / DeviceMotionEvent browser compatibility

1 Remap controls and verify two output identities

Establish the ordinary controller path before evaluating a specialized capability. The output identity and the physical source are separate choices.

Equipment and preconditions

- A recognized gamepad, Windows, PadForge, and a controller-aware game or input viewer.
- Use a loaded review profile, a running engine, and the installation checks above. Keep a record of the recipient's original bindings.

Procedure

1. On Dashboard, choose Add Controller > Xbox. On Devices, assign the physical gamepad to that slot.
2. On Mappings, record physical B on output A, then physical A on output B. Press and release each separately.
3. Compare Devices input, the mapped Value, and the recipient's own response. A Preview animation alone does not establish the last step.
4. Create a separate PlayStation slot and select a specific DualShock 4 preset before assigning the source. Then assign it and use Mappings > Copy From to translate the Xbox mapping.
5. Enable only the output being checked. Reopen the recipient if it caches controllers, then repeat the two-button check for the PlayStation identity.
6. Optionally bind an (Any Device) gamepad role and substitute another recognized model. Repeat both press and release checks before calling the mapping portable.

Result and evidence

A passing observation has one intended recipient action per press, a clean release, and the selected output identity. Prompt artwork is a separate game choice and should be reported only if observed.

- Keep the selected preset label, physical device name, mapping source/target, and the recipient result together in the result record.
- Record whether the recipient needed a restart or rebinding after the identity changed.

Limits and reset

- Copy From leaves destination macros alone. It is not the same operation as a full mapping Paste.
- Any Device gamepad roles require a recognized gamepad layout. Raw joysticks and specialized named-input rows have different rules.

- Reset by disabling the trial slots, restoring the previous bindings and hiding choices, and loading the preserved original profile.

Sources: Xbox / Xbox slots / Button pressure

2 Combine devices and check the final output

Use one output row to expose how several inputs combine, then distinguish its intermediate Value from the recipient's final output.

Equipment and preconditions

- Two assigned input sources, such as two gamepads or a gamepad and a keyboard, and an input viewer.
- Use a disposable profile and a recipient that cannot trigger unrelated desktop actions. Pick two button-like sources the reviewer can operate independently.

Procedure

1. Assign both sources to one Xbox slot. Choose an otherwise unused output button, bind the first source, and add the second with + Add Source.
2. Set Combine to Either. Exercise first-only, second-only, both, and neither. Record all four states.
3. Set Combine to Both and repeat the same sequence. Check that releasing either source releases the combined output.
4. For an analog comparison, bind two stick-axis sources to an axis row. Compare Strongest with Average using a held half-deflection on just one source.
5. For a separate SOCD check, create a Keyboard + Mouse slot and assign the gamepad. Map two buttons to opposite keyboard-output directions, then configure that slot's Output SOCD key pair and mode. Repeat both press orders and both release orders in the receiving application.
6. Unassign one source and check that the remaining source still follows the intended rule. Restore the assignment before saving.

Result and evidence

The button truth table should match Either and Both. The analog recipient should expose the chosen combine rule. Judge SOCD after the final output reaches its consumer, because the mapping Value precedes final SOCD processing.

- Record the four button states, axis positions and chosen rule.
- Keep the recipient result separate from raw input and mapping Value.

Limits and reset

- Combined values can clamp at full scale. Sensor rows have a narrower combine-mode set.
- A valid SOCD configuration is not a statement about a game's or tournament's current rules.
- Remove the extra source or reload the preserved profile to return to the original mapping.

Sources: Auto-mapping / Multi-slot device assignment

3 Compare analog depth with Rapid Trigger

Separate continuous key or trigger depth from the movement-based digital actuation produced by Rapid Trigger.

Equipment and preconditions

- A supported analog keyboard or a gamepad trigger that reports depth, and a viewer that displays a trigger or button output.
- For a keyboard, enable Read Analog Keyboards and select its named row. Match the exact model and firmware prerequisites. Razer analog models need Synapse. Most competing configurators must otherwise release the shared channel.

Procedure

1. Press the source slowly and inspect its own depth display. Confirm that partial values exist before proceeding.
2. Bind the source directly to an analog trigger output. Check shallow, middle, full, and released positions in the recipient.
3. Use a second test row that supports a press, choose Primary Mode > Rapid Trigger, set an example actuation threshold of 20% and Distance 10%. These are review settings, not a universal calibration.
4. Move above 20% to actuate. From a deepest reading near 40%, lift to about 35%, then to about 28%. The five-point lift should not release, while the twelve-point lift should.
5. From the released low point near 28%, push to about 32%, then about 40%. The four-point push should remain released, while the twelve-point push should reactivate.
6. Return below actuation, then repeat. Record actual source values rather than claiming exact hand travel from the example numbers.

Result and evidence

Direct mapping should retain intermediate depth. Rapid Trigger should produce distinct transitions from sufficient travel and reset below actuation. A direction reversal alone is not a passing actuation result.

- Record source depth beside the recipient state for every transition.
- For a keyboard, record whether normal typing also reached the focused application.

Limits and reset

- One Distance serves press and release. Continuous Rapid Trigger and position-row outputs are not offered by this mode.
- The release documentation lists analog-keyboard hardware testing as unperformed. A populated device row is not a key-depth test.
- Return the row to Direct or reload the profile. Restore the original vendor-software state.

Sources: Rapid Trigger / Turning it on / PadForge 5.0.0 release notes

4 Calibrate gyro and combine it with Flick Stick

Use a measured game turn to connect physical motion and stick direction to a mouse-driven recipient.

Equipment and preconditions

- A working gyro controller and a game that accepts mouse aim. Use an offline practice scene or another controlled mouse recipient.
- Create a Keyboard + Mouse slot, assign the pad, and preserve the game sensitivity. Choose the grip that matches how the device is held.

Procedure

1. Open Gyro, place the controller still, and run Calibrate Gyro. Record the status and at-rest readout after completion.
2. Map Gyro Pitch to Mouse Y and Flick Stick (Right Stick) to Mouse X.
3. Start with an explicit example: Dots per 360 degrees 14400, Flick Time 0.1 s, Flick Threshold 0.9, and 90-degree snap. Keep vertical sensitivity at 1.0x initially.
4. Flick right and compare the intended quarter turn with the actual game view. Adjust Dots per 360 degrees against the recipient, then keep game sensitivity fixed.
5. Sweep the stick around its rim and release. Check horizontal turning, then test vertical gyro movement with the stick resting.
6. Add an engage input if desired and verify both engaged and disengaged motion. Repeat after changing grip only if that is part of the review.

Result and evidence

A passing result has reproducible horizontal turns and separately controlled vertical motion. Releasing Flick Stick leaves the camera where it turned. Gyro movement stops when its configured gate is closed.

- Record controller transport, calibration state, grip, mouse sensitivity, Dots per 360 degrees, and the observed turn.
- Report at-rest drift from the recipient and displayed sensor values separately.

Limits and reset

- Flick Stick supplies horizontal turns only. Example calibration values are not valid for every game.
- A held stick entering a layer need not flick unless Allow Flick on Engage is enabled.
- Restore game sensitivity and the original mappings. Mark the vertical portion skipped if no gyro source is available.

Sources: Flick Stick / Output routing and rate versus tilt

5 Generate motion and distinguish native reports from DSU

A sensorless source can generate motion, but the selected virtual preset and the receiving API still determine what reaches the application.

Equipment and preconditions

- A gamepad or keyboard source and a motion-aware viewer, emulator, or compatible game. A DSU-capable recipient is useful for a separate route check.
- Use a PlayStation slot with an explicit DualShock 4 preset for the first native-output attempt. For DSU, use slot 1 through 4 and a local consumer configured for the server port.
- The Nintendo picker offers four presets: Nintendo Switch Pro Controller, Nintendo Switch Pro Controller (USB), Nintendo Switch 2 Pro Controller and Nintendo Switch 2 Pro Controller: Full.

Procedure

1. Assign an online source. Bind Right Stick Y to Motion Pitch and choose Speed.
2. Set example values of Top Speed 180 degrees/s, Start Speed 0, and Motion Deadzone 20%. Hold a deflection and release it while observing the recipient's motion display.

3. Change Motion Pitch to Angle, set an example Lean Angle of 30 degrees, and repeat. Check held lean and return toward level on release.
4. If a physical sensor is also assigned, record the physical-only and combined cases separately. Since 5.1.0 an emptied motion row switches that channel off, and a layer with no motion row of its own uses Base.
5. For DSU, enable the server and bind the consumer to the intended local slot. Record gyro/acceleration reception independently of the virtual gamepad buttons.
6. If comparing presets, repeat on plain DS3, plain Switch 2 Pro, or plain Steam Deck only as a DSU case. Their native reports do not carry the configured motion. Switch 2 Pro: Full does carry it, and only Steam and SDL built with libusb read that preset.

Result and evidence

Speed should sustain a turn rate while held and stop that rate on release. Angle should hold pitch or roll lean and level on release. DSU reception establishes that separate feed, not motion in the HID report.

- Record source, mode, parameter values, output preset, recipient backend, DSU slot if used, and the actual motion response.

Limits and reset

- Yaw has no Angle mode. Full DS3 has only accelerometer and yaw gyro, with the documented pitch/roll tilt restrictions.
- The docs settle the empty-row case since 5.1.0: an emptied row sends nothing, and a row whose devices are all offline hands the channel to another live row for the same sensor. Neither release reports a game-side simulated-motion test.
- Disable the DSU listener if the trial enabled it and restore the original motion rows and recipient bindings.

Sources: Motion Pitch, Yaw and Roll / Source selection / Motion Snapshots / Motion Passthrough / Nintendo / DSU source selection

6 Open a layer menu and fire a scoped macro

Build a complete input-to-layer-to-menu-to-macro path with an explicit scope and a visible receiving result.

Equipment and preconditions

- A gamepad, a Keyboard + Mouse slot, and an empty local text document.
- Use a named review profile. Keep the text document focused when the action fires.

Procedure

1. On Mappings, create a Tools shift layer with left bumper as its button activator, Mode Hold, Delay 0, and Also Fire Activator's Own Mapping off.
2. On Macros, add Review text. Set Fire to On Press, Layer to Any Layer, and the repeat behavior to once. Add a Text Block action containing a short neutral phrase. Leave the macro's own input trigger empty.
3. On Menus, add a two-cell Radial Ring. Set Layer to Tools and turn on Layer Holds the Menu Open.

4. Choose Right Stick under Steer With. Use On Click and the host's default stick click. Bind one cell to the named macro and the other to Backspace.
5. Hold left bumper, steer to the macro cell, click once, release the click, and then release the bumper.
6. Check that the text appears once, the menu closes, and ordinary Base input works afterward. Repeat while another existing layer is active only if that interaction is part of the review.

Result and evidence

The held Tools layer should open the menu, the click should trigger the named macro, and release should end the intended held state. Record the phrase that appears in the text document. A menu highlight alone does not show that the macro ran.

- Keep the activator mode, menu fire mode, macro Layer setting, and output document result.
- Record any unintended extra key or repeated action.

Limits and reset

- Base macros can run under a layer that inherits unmapped Base targets. Any Layer avoids that ambiguity in this example.
- Since 5.1.0 an On Touch Release commit made by leaving a layer runs a macro scoped to that layer, as the shift-layers guide says. Check that case separately before relying on it. Other activators or a Latch/Cycle override can prevent a Hold release from returning to Base.
- Release every activator and reload the review profile if a state remains engaged. Delete the disposable output text.

Sources: Triggers / Activator dialog and modes / Style, host, fire mode and cells / The ShiftActivator DTO

7 Check profile ownership and foreground selection

Verify what changes with a profile, what remains global, and which profile is actually active after focus or restart.

Equipment and preconditions

- A working manual layout and two harmless applications with identifiable executables.
- Preserve Default and create a separate trial profile. If starting from the gallery, assign the controller to every slot that the starter creates.

Procedure

1. Make a visible mapping difference and use Save As to create a named profile. Load it explicitly and inspect its slots and bindings before using automatic selection.
2. Add the actual game/application executable and enable Auto-Switch Profiles Based on Foreground Application.
3. Switch focus between the matched application and an unmatched application. Record the FOREGROUND line and the active profile name in both states.
4. Manually load the trial profile and check its behavior separately. If a launcher has set an external hold, manually load Default to clear it before evaluating automatic switching.
5. If reviewing the Razer Chroma, Logitech LIGHTSYNC or Razer Sensa rows, assign one to a virtual controller while the named profile is active, then switch profiles and check that the assignment

follows the profile. A profile saved by 5.0.0 turns its switch opinions into assignments once, when it loads.
6. Fully exit and relaunch PadForge. Record the final active profile and receiving output. Default can load as a baseline before the named shutdown profile is restored.

Result and evidence

A passing result distinguishes automatic selection, manual selection, startup restoration, and a vendor-row assignment. Edits must land in the intended active profile rather than merely in the card selected for editing.

- Record executable rule, active-profile indicator, the changed mapping, and restart outcome.
- For a vendor row, record its assignment in each profile before and after the switch.

Limits and reset

- A legacy profile missing the macro field leaves current macros in place.
- Changing identity, enabled state, or slot topology can recreate or remove output. The first-instance external hold with its pipe off can survive an engine stop.
- Restore the original profile, vendor-row assignments, and auto-switch setting. Keep the exported trial profile private until its paths and assets are checked.

Sources: Contents and auto-switch / The command-line form / Profile Loading / Default Profile Snapshot Mechanism / The old Dashboard switches / Complete Property Reference

8 Translate a Steam layout and inspect the losses

Evaluate a translated configuration as a collection of supported bindings, not as an automatic whole-layout conversion.

Equipment and preconditions

- Internet access, a controller, and an owned game for which Steam returns a compatible community configuration.
- Read the Community Configs disclosure in Settings. Besides the search text and the game, config and creator IDs, the anonymous Steam logon sends the OS type, a language and a hashed machine ID. Legacy configurations can have additional Steam requirements.

Procedure

1. Enable Community Configs in Settings and open Profiles > Browse Community Configs.
2. Search the game, select a controller type, and open an actual returned configuration. Record its real title and creator locally.
3. Read Clean, Partial, and Skipped, then inspect the reason for every non-clean row that matters to the game.
4. Use Save Profile first. Load the new profile deliberately, assign the source to the created slots, and inspect mappings, macros, menus, and action-set layers.
5. Choose one clean ordinary binding, one layer/menu action, and one partially translated behavior. Compare the declared result with the receiving game.
6. Record that game's Steam Input override. Isolate the PadForge route for the initial comparison, then use the publisher's game-specific guidance where relevant.

Result and evidence

The imported profile should expose editable supported bindings and a manifest of reported translation limits. A useful result identifies which selected behaviors work, which differ, and which still require Steam Input or manual recreation.

- Keep the real manifest totals and specific source-to-target examples.
- Record any native game action with no equivalent keyboard or gamepad binding.

Limits and reset

- Some dropped settings receive no manifest note. Clean does not establish complete parity.
- No universal Steam Input-off rule follows for all games or adaptive triggers. Publisher conditions and the virtual output path need separate evaluation.
- Restore the original Steam layout and per-game override, exit PadForge through the tray if reverting its outputs, and retain the import for later inspection.

Sources: Opt in / Enable Community Configs / Clients

9 Separate physical DualSense effects from game-authored feedback

Use a manual effect as a physical-path check, then treat a game-generated effect as a separate result.

Equipment and preconditions

- A physical DualSense or DualSense Edge, preferably USB for the initial comparison, and an input/feedback-aware game for the optional second phase.
- Select the physical pad in the slot's device picker. Record its transport and the virtual preset independently. Use low, deliberate effect settings.

Procedure

1. With no game writing trigger effects, open Adaptive Triggers. Set one trigger to Weapon with example start 100, end 180, and strength 100. Leave the other Off.
2. Pull slowly through the configured region and record whether physical resistance occurs. A preview curve or marker alone is insufficient.
3. Reset the manual effect. For a game phase, choose an explicitly documented PC edition and retain its published USB/Bluetooth and Steam Input conditions.
4. Record the selected virtual Sony preset. A Full preset exposes a different virtual configuration from an ordinary HID preset, and neither name proves game acceptance.
5. Trigger a repeatable in-game event and distinguish ordinary rumble, adaptive resistance/vibration, haptic audio, and speaker sound.
6. If examining trigger-state readback, use a USB-form DualSense or Edge virtual preset and a real DualSense-family input. Keep that input requirement separate from its physical cable/radio choice.

These publisher examples describe physical-controller prerequisites. They are starting conditions for an owned PC game, not evidence that PadForge was accepted.

PC edition	Published condition	Boundary
Returnal	The PC page requires a wired connection for the advertised full DualSense feature set.	No title-specific Steam Input rule or PadForge runtime result is established here.
Marvel's Spider-Man: Miles Morales	Patch 2.516.0.0 documents adaptive triggers with Steam Input, while full speaker/haptic feedback remains wired.	The publisher announcement is June 1, 2023. Later help-center timestamps are not that patch date.
Helldivers 2	The developer FAQ requires Steam Input off before launch for the documented PC haptics.	The FAQ does not supply a title-specific USB/Bluetooth feature split.

Result and evidence

A manual physical effect verifies that one output path can drive the pad. A game result additionally needs the actual recipient event and response. Record each feedback channel separately.

- Keep game edition/build, preset, physical model/transport, Steam Input setting, game settings, and physical outcome.
- Treat a publisher's controller requirements as prerequisites, not as a PadForge compatibility certification.

Limits and reset

- Game writes take precedence over the tab. Xbox impulse motors cannot reproduce adaptive resistance effects.
- Published physical-controller support does not establish virtual-device discovery or controller-audio forwarding through PadForge.
- Return both triggers to Off/Reset All, close the test game, and restore the original output preset and game override.

Sources: Returnal PC / Miles Morales patch 2.516 / Helldivers 2 FAQ / Availability and modes / PlayStation

10 Trace controller audio and its processing route

Follow a known audio source into a physical speaker or actuator route without treating every controller as the same audio device.

Equipment and preconditions

- A supported Sony speaker route, or a separately documented haptic-tone/Steam Controller route, and a local audio clip on a named Windows output.
- Begin with a quiet source. DualSense/Edge audio is documented over USB and Bluetooth. DS4 reaches its speaker over Bluetooth only. On a cable the second model, and any DS4 on the Sony wireless USB adapter, plays through its headphone jack, and a first-model DS4 on a cable has no audio.
- Crossfeed needs a real stereo headset route. Mono and Speaker Only routes skip it.

Procedure

1. Select the physical pad on the slot. Enable Mirror System Audio to the Controller Speaker and select the actual Mirror Source. Master Volume 20 is an example for DualSense/Edge or a haptic-tone path. On a DS4 or Wii Remote it sets the level of macro sounds only.
2. Play the clip and listen at the physical destination. Stop the mirror and verify the sound stops. An output endpoint can include several programs.
3. On a supported Sony PCM path, compare crossfeed, one parametric EQ band, and the limiter one at a time. Record the controls and audible result without calling the displayed curve an acoustic measurement.
4. If importing AutoEq, use a parametric profile and record the resulting bands/preamp. Graphic EQ files are not the documented import format.
5. For a haptic-tone pad, compare High Tones and the mirror gate using the same clip. Label it a dominant-tone approximation, not speaker reproduction.
6. For Steam Controller 2026, keep USB/dongle audio streaming separate from Bluetooth single-tone behavior. Do not transfer a result between those transports.

Result and evidence

A passing result identifies the physical output, its source and transport, and whether the chosen control changes that output. Master Volume is slot-wide. On a DS4 or Wii Remote it sets the level of macro sounds only. Headphone Volume, on a DualSense and on a DS4 over Bluetooth, is a separate jack control.

- Record model, transport, mirror endpoint, volume controls, DSP changes, and the audible or tactile result.
- If measuring delay, document the timing method and reference. Buffer numbers are not milliseconds or a measured end-to-end result.

Limits and reset

- The DualSense Bluetooth buffer affects speaker, haptics, and microphone together.
- Wired DualSense audio can drop a stale Bluetooth link once when its path is built. That is separate from the charging-state Quick Charge option.
- Disable mirroring, restore DSP and volume settings, and stop any test sound.

Sources: Availability and mirror / Power

11 Receive directional wheel force and route feedback to a shaker

Compare the game's virtual force interface with the physical wheel path, then inspect the separate rumble-to-audio output.

Equipment and preconditions

- A supported wheel, a DirectInput force-feedback recipient, and optionally a wired audio output, amplifier, and tactile transducer.
- Mount the wheel and start with low force. Use the exact vendor dependency for its model. Thrustmaster needs its vendor driver. Preserve game bindings.

Procedure

1. Create an Extended slot with an appropriate force-feedback preset and assign the wheel. Inspect Clone Device 1:1 confirmation, then bind remaining axes/buttons manually.
2. If Wheel exposes Rotation Range, choose an explicit value such as 900 degrees. Set Auto Centering Strength to 0% if present. Skip controls the selected model does not expose.
3. Set Force Feedback Overall Gain to an example 30%. Bind the virtual wheel in the game and use a controlled practice drive.
4. Record steering response, a directional force event, and release back toward idle. Separate game-authored force from an idle centering override.
5. If a shaker is available, enable Route Rumble to an Audio Output on Bass Shakers, choose its dedicated endpoint, Mono (All Channels), Master Gain 20%, and example Low/High frequencies of 40/80 Hz.
6. Use Test to verify the audio path, then trigger real game feedback and inspect the live meters and physical response. Compare the two results explicitly.

Result and evidence

A passing wheel result identifies the direction and timing of game force at the physical wheel. A shaker result needs an actual tactile output driven by received game feedback. A Test tone alone verifies only the audio route.

- Record game build, recipient API, output preset, wheel model/driver, gains, and force event.
- For shakers, record endpoint, channel mode, source voice, and real-event meter/physical response.

Limits and reset

- A nonempty wheel descriptor does not guarantee every force effect. An Xbox slot supplies rumble rather than native DirectInput wheel force.
- Steering Angle Rumble is vibration, not torque. Bass Shakers does not replace SimHub telemetry, dashboards, or effect formulas.
- Disable the shaker route, release manual forces, and restore the original game and wheel settings.

Sources: Clone Device 1:1 / Directional, condition, and Extended force feedback / Native routing / Routing, voices, and sweep

12 Use MIDI as input and then as output

Test the two MIDI directions independently so a working piano preview is not confused with a working virtual endpoint.

Equipment and preconditions

- A MIDI source, a MIDI monitor or DAW, and a MIDI path that starts: Windows MIDI Services through the API built into Windows or the runtime PadForge installs on Windows 11 24H2 or later, or the legacy MIDI API on any Windows. For output under the legacy API, an existing port such as a loopback driver.
- Read the Windows MIDI Services card in Settings and record which path it names. Under the legacy API, assign the MIDI source to a slot before expecting its preview to move, since that API opens a port only while a slot has it.

Procedure

1. For input, select the named MIDI device on Devices. Use the piano and CC display to identify note 60 and a known CC such as CC1.
2. Assign that device to an Xbox slot. Bind the note to a button and the CC to a trigger. Check note hold/release and the analog CC range in the recipient.
3. If using an endless encoder, record its configured encoding. Confirm a detent in each direction produces the intended separate pulse before using it to change a value.
4. For output, add a separate MIDI slot and assign a gamepad. Set example Channel 1, CC Count 1, Start CC 1, Note Count 1, Start Note 60, and Velocity 100.
5. Bind a face button to the note row and a trigger to the CC row. Under Windows MIDI Services, select the PadForge MIDI endpoint as an input in the monitor or DAW. Under the legacy API, pick the existing port as the slot's Output Port and select that port's other end in the receiver.
6. Record Note On, Note Off after release, and the full CC range. If the endpoint is visible but messages are absent, separate slot/engine state from receiving-app input configuration.

Result and evidence

A passing input result has the named source and the mapped controller response. A passing output result has actual MIDI messages, channel and values in the receiver. Endpoint visibility alone is insufficient.

- Record runtime status, endpoint names without private identifiers, channel, note/CC numbers, encoder mode, and press/release messages.

Limits and reset

- Input note velocity is not an analog source, and channels collapse onto the same note or CC. Channel pressure, polyphonic aftertouch, and program changes are not mapped.
- Output Velocity is a configured note value, not the physical depth of the button. Two's-complement or signed-bit encoder modes do not establish the documented binary-offset relative path.
- Disable or remove the trial slots and restore the DAW input selection. Do not uninstall a shared runtime for rollback.

Sources: Requirements / MIDI / Microsoft MIDI releases / What's required vs optional / Microsoft MIDI artifact removal / Installing the MIDI runtime

13 Turn a phrase or NFC tag into a named action

Use explicit named inputs to compare a short recognition pulse with a tag that can remain present.

Equipment and preconditions

- A system microphone, a supported DualSense microphone path, or a headset in a Bluetooth DualShock 4's jack. For the optional tag branch, a compatible PC/SC reader or original Switch Pro/right Joy-Con over Bluetooth and a test tag.
- Use a disposable text output or a named Toggle layer. Select specialized inputs by name rather than relying on Any Device.

Procedure

1. From the microphone row, open Manage Voice Macros. Enable Voice Macros, set Listening to Always, and start with Minimum Confidence 0.80.
2. Register a short phrase with a neutral label. Assign the microphone to the test slot and use that named phrase as a Toggle-layer activator.
3. Speak once to enter the layer and again to leave. Record recognition text/confidence separately from the fired phrase indicator.
4. For NFC, select the reader or Switch controller, open Register / Manage NFC Tags, tap a test tag, name it, and register it.
5. Assign the reader to the slot and bind the named tag to a harmless action. Compare a tap, a held tag, removal, and a second tap.
6. If using a Switch reader, keep its Bluetooth requirement and shared IR/NFC resource use in the result record. Do not present this as game-specific Amiibo emulation.

Result and evidence

A voice result should show the 175 ms phrase pulse producing the intended persistent Toggle change. A tag result should show the correct named source and receiving action. Recognition confidence alone is not proof the configured action fired.

- Record microphone path, phrase label, threshold, recognition and action outcome.
- For tags, record reader type and timing without publishing the tag UID.

Limits and reset

- Voice uses an en-US model and can miss phrases. A phrase is not a held voice key.
- PC/SC taps pulse briefly. A tag resting on an original Switch reader remains held and has a longer removal delay. Switch 2 NFC is not supported by this path.
- Disable listening, release the layer, remove trial bindings, and keep tag identifiers out of shared profiles or reports.

Sources: Microphone paths / Reader paths and timing / Trigger source / Trigger Detection / Gamepad sources

14 Use a phone for controls and a separate menu surface

Compare touchscreen input, browser-exposed gamepad input, and Web Menus without treating them as one interchangeable device.

Equipment and preconditions

- A phone/tablet and a reachable PC on the local network. For Browser Gamepad, a controller already paired to the browser device.
- Enable Web Controller Server on Dashboard and use its displayed address. Record browser/platform/version. Use HTTPS for phone motion and wake-lock eligibility.

Procedure

1. Open the address and choose Xbox 360. Assign its web-controller row on Devices to a test slot. Check a button, a stick, and release in the recipient.
2. For Browser Gamepad, choose that layout, press a button on the paired pad so the browser exposes it, and assign its separate input row.

3. Record whether the browser supplies a standard or raw layout. Check its actual buttons and axes rather than assuming every physical control is exposed.
4. For motion, use a gamepad layout's Motion button and grant permission where required. Verify actual sensor values before trying a gyro mapping.
5. For a command surface, create a Touch Grid menu, enable Show on Web Controller, choose Web Menus on the phone, and assign it to the slot. Tap a cell bound to a harmless local action.
6. Release or close the page during a held input and verify the destination eventually releases it. Record the observed interval and reconnect behavior.

Result and evidence

A passing result distinguishes a touch/controller source from a menu-cell trigger and records the PC's receiving action. A loaded page or visible Motion button is not a sensor or haptic result.

- Record browser version, layout, reported control counts, permission outcome, server route, and recipient release behavior.
- Omit access codes, QR contents, and private network addresses from shared evidence.

Limits and reset

- Web Menus acts on its host PC and is not shared over Remote Link. iPhone Safari has no gamepad haptics or phone vibration in the reviewed sources. Android Firefox does not produce phone vibration through Navigator.vibrate despite a possible success return.
- Plain HTTP has different sensor and wake-lock limits. Gamepad API access has no universal current HTTPS requirement. Session restore can retain an access cookie.
- Stop the listener if enabled only for the trial and use New Code to revoke the optional plain-address code when needed.

Sources: Setup and network / Assign devices from the other PC / Session cookies / Motion permission / Vibration API browser compatibility

15 Share input and returning feedback between two PCs

Test the owner-to-consumer input path and the returning feedback path as distinct directions.

Equipment and preconditions

- Two PCs running PadForge, a supported physical gamepad on the owner, and a game/input viewer on the consumer.
- Begin on a controlled local network. Record both builds. Choose the pairing restriction deliberately and preserve existing assignments.
- Read what Remote Link shares before pairing. Discovery broadcasts the PC's name, identity fingerprint, protocol version and link port on the local network, and the internet path uses public STUN servers, the public BitTorrent DHT and n0's iroh relays.

Procedure

1. Enable Remote Link on both PCs. Choose the nearby peer, compare the six-digit pairing code on both screens, and confirm both ends.
2. On the consumer, assign the owner's shared device to an existing virtual slot. Check one button and one axis in the receiving application.

3. Trigger a known game rumble event on the consumer and observe whether it reaches the owner's physical pad. Record the direction explicitly.
4. To evaluate remote assignment, grant Allow Remote Assignment Changes on the consumer. From the owner, use Assign Shared Devices to select existing destination slots.
5. Change one assignment, verify the consumer's actual state, then revoke the permission and check that future changes are blocked while existing assignments remain.
6. Stop the source or disconnect the peer during a held input. Record release and reconnect behavior before testing an internet-code route separately.

Result and evidence

A passing result has input at the consumer and the chosen feedback at the owner. Pairing, discovery, or a missing reply alone does not establish the final assignment or feedback result.

- Record owner/consumer roles, two builds, physical transport, output preset, recipient, permission/restriction state, and each direction's result.
- If a request times out, refresh the assignment view rather than assuming it failed.

Limits and reset

- This is input/effect transport, not arbitrary USB forwarding. Web Menus is excluded, and received devices are not relayed onward.
- Gamepad-only pairing suppresses specified keyboard/mouse and output paths, but Run Program and MIDI still operate. It is not a general sandbox. New 5.0.0 device data lacked a two-PC maintainer bench at release.
- Restore assignments and permission settings, revoke the test pairing if it is no longer needed, and stop the trial listeners.

Sources: Input and feedback / Profiles from the phone / Web Menus (#471) / Discovery / What Remote Link shares

16 Check retro pressure and a VMU display or Pak backup

Keep analog pressure, accessory display, and save storage as separate capabilities with different evidence.

Equipment and preconditions

- A supported Bliss-Box adapter and one available accessory path: DualShock 2, Dreamcast controller with VMU, or N64 controller with Controller Pak.
- Close competing Bliss-Box utilities. Match the adapter firmware to the selected route. Use a separate destination filename for a pak backup.

Procedure

1. Enable Read Bliss-Box Adapters. Select the port on Devices and record the detected controller and firmware. Reassign an already mapped port if enabling the reader changes its raw layout.
2. For DualShock 2, inspect the named pressure sources. Bind L2 Pressure and R2 Pressure explicitly to trigger pull instead of leaving their digital button bindings.
3. Compare shallow and full pressure in the source display and receiving trigger value. If using Full DS3 output, treat its additional pressure rows and game acceptance as a separate attempt.

4. For VMU, choose Dreamcast Screen and a static supplied image. Compare the converted preview with the physical display after the write finishes.
5. For Controller Pak on firmware 3.0 or later, use Back Up Controller Pak and save to a new file. Preserve it unchanged and inspect its size. Do not perform a restore for this review route.
6. Record only the branches actually performed and identify absent accessories as skipped.

Result and evidence

A pressure result needs intermediate values at the intended output. A VMU result needs the physical static image. A pak backup needs the saved 32 KB MPK file. A progress dialog alone does not show a backup.

- Record adapter model, firmware, detected controller, source names, output preset, and each accessory outcome.
- Keep save contents and user identifiers private.

Limits and reset

- The documentation reports no live Bliss-Box bench for these new routes. Full DS3 pressure had not reached a maintainer-tested game at release.
- VMU updates are slow, can hold input briefly, and are not smooth video. Restore replaces the entire target pak and is outside this read-only storage check.
- Restore the original assignments and reader state. Keep the backup separate from the review report.

Sources: Turning it on / PadForge 5.0.0 release notes

17 Keep head-tracking input separate from virtual VR output

Use one real pose input route and, only when available, a separate SteamVR controller-output route.

Equipment and preconditions

- For pose input, an existing OpenTrack/FreeTrack setup or a tracked headset with a compatible OpenXR runtime. A Quest on Link or Air Link needs SteamVR's runtime. For virtual output, an installed SteamVR runtime and an accepting consumer.
- Preserve the tracker's calibration and the game's current input route. A TrackIR/NPClient-only game is not automatically a PadForge gamepad consumer.

Procedure

1. For same-PC OpenTrack UDP, enable UDP Tracking Input and direct OpenTrack to the local address and selected port, default 4242. Keep only one listener on that port.
2. Select the named Head Tracker row and move yaw, pitch, and translation one at a time. Verify fresh values before binding an axis to a controller output.
3. Set explicit rotation/translation ranges and check full deflection in the recipient. Record signs rather than assuming a camera tracker's axes match the game.
4. For OpenXR, select the intended runtime, choosing SteamVR for a Quest, enable headset input, and verify actual pose/control samples. Set Neutral applies to the OpenXR headset and hand poses together.

5. For a separate virtual-output attempt, add the one VR slot, assign a gamepad, and map a stick, trigger, and grip for each hand. Inspect Preview, then the actual SteamVR consumer.
6. Pause tracking or let a physical controller sleep and record which axes/controls return to rest. Do not infer live input from a device row alone.

Result and evidence

A head-tracking result needs fresh pose and a receiving response. A virtual VR result needs the consumer to accept the mapped hand controls. The virtual hands are anchored relative to the headset rather than supplied with room-position tracking by PadForge.

- Record runtime, physical source, input API, output API, axis sign/range, sample-loss behavior, and the recipient result.

Limits and reset

- OpenXR needs a runtime that permits the background session, which Meta's runtime does not offer in any version Khronos lists. On a runtime other than SteamVR, PadForge restarts a session the runtime loses, as it has since 5.1.0. ARM64 needs a matching vendor runtime.
- OpenTrack tracking and filters remain upstream. One VR slot supplies both hands. Preview and SteamVR's own Test Controller are not interchangeable with a game result.
- Disable the trial input and VR slot, restore the original tracker output, and leave shared runtimes installed.

Sources: Inputs and ranges / Windows on ARM / Turning it on / Consuming real VR devices / VR / OpenXR runtime for a Quest

18 Check vendor controls and feedback without claiming full replacement

Use a named vendor path to expose its dependency, then separate that input or feedback from the specialist software's other functions.

Equipment and preconditions

- One available route: Logitech G-keys, a handheld hidden button, a Razer, Logitech or SteelSeries mouse or keyboard PadForge can light, a haptic mouse, or a vendor row with its software installed.
- Record the exact hardware, host version, and existing profile. Do not equate a lighting engine with the G-key SDK or install unrelated packages to force a result.
- PadForge lights a device on its own while its vendor's software is closed, and through Synapse, G HUB, Logitech Gaming Software or GG while that runs. The Razer Sensa row needs x64 PadForge, Synapse 4, Sensa hardware, and the Sensa HD Games haptic source. A Rival needs SteelSeries GG. An iFeel mouse needs Play Rumble on This Mouse, which takes it away from Windows as a pointer until Restore the Windows Driver.

Procedure

1. For G-keys, enable Read Logitech G-Keys. In Logitech Gaming Software 8.55 or later, drag the G-key command onto each key in the PadForge profile and make that profile persistent. Take a G600 or G300 out of on-board mode.
2. Select the named G-key row and bind one key to a harmless output. Compare the PadForge window focused with the intended recipient focused. Record both results rather than promising background callbacks.

3. For a hidden handheld button, enable Handheld PC Buttons, select Hidden Buttons, and use Learn / Manage Hidden Buttons. Follow the idle period, press/release window, and Register step.
4. Verify the learned named source and its output with vendor tools in the same state used during play. Check whether the vendor action also occurs.
5. For lighting, assign the mouse, keyboard or vendor row to a virtual controller, check Control This Device's Lighting on a mouse or keyboard, and pick a mode. Read the route line, then observe the device with its vendor software closed and again with it running. On a virtual DualSense or DualShock 4, check that a game's lightbar write takes over.
6. For rumble, assign a haptic mouse or the Razer Sensa row and use Test Rumble at a low, a middle and a full level. Record the route line and, for Sensa, the configured haptic source.
7. Repeat after disabling the trial feature to distinguish its contribution from vendor software still running.

Result and evidence

A passing result identifies a particular input callback, learned control, or physical light or haptic response. The presence of an SDK file, a fixed 102-entry source row, a Lighting tab, or a selected vendor option is only a setup observation.

- Record dependency status, exact model, vendor software state, route line, foreground/background state, source/output pair, and any duplicate vendor action.

Limits and reset

- G HUB does not include the G-key SDK, so G-keys need Logitech Gaming Software. Logitech's manual and two other implementations indicate that a persistent profile carries key events to an application in the background. PadForge's own G-key path has not been run on Logitech hardware.
- Handheld input learning does not replace TDP, GPU, display, or firmware mode controls. The Razer Sensa row plays one strength, not stereo haptics. The 5.1.0 release notes say none of the mouse and keyboard lighting or rumble hardware has run on the bench.
- Software that lights every device of a kind in one color gives the color to the lower-numbered virtual controller. Some Razer and SteelSeries models keep the last color after PadForge lets go.
- Restore the vendor profile, the PadForge assignments and lighting checkboxes, and put an iFeel mouse back with Restore the Windows Driver. Remove only the trial bindings or learned entries, without changing shared drivers.

Sources: Turning it on and requirements / Turning it on and learning / Logitech G-key SDK 8.57.148 / Windows on ARM / Setting one up / Vendor rows / Haptic mice and the Razer Sensa row

19 Type and point through a keyboard or mouse model

A Keyboard + Mouse slot sends through Windows input or through a HIDMaestro keyboard or mouse model that programs see as that hardware. Check what the model reports and what the recipient receives.

Equipment and preconditions

- A gamepad source, a disposable text document, and a harmless program that reads mouse input. Device Manager, to see the virtual device.
- Use a separate Keyboard + Mouse slot. HIDMaestro installs itself the first time a half needs a model.

Procedure

1. Add a Keyboard + Mouse slot, assign the gamepad, and bind a face button to A and the right stick to mouse movement. Confirm both through Windows Input first.
2. In the bar above the slot's tabs, pick a keyboard model under Keyboard. Read its tags: Keys at Once, Media Keys and Lock Lights.
3. Type the bound key into the text document. Check that the rows the model cannot send are hidden, and that they come back with their bindings when the half returns to Windows Input.
4. Pick a mouse model under Mouse and repeat with pointer movement, a button and the wheel. While the slot runs, find the model's keyboard or mouse in Device Manager.
5. On a model with lock lights, toggle Caps Lock on the physical keyboard and check that the preview shows the lock light Windows sets.
6. Turn the slot off and check that the virtual device leaves Device Manager and that no key stays held.

Result and evidence

A passing result shows the typed character or the pointer motion in the recipient through the model, with the model listed as a device while the slot runs. The preview drawing alone is not the output.

- Record the model names, their tags, the Windows input language, the recipient, and the device listing.

Limits and reset

- The 5.1.0 release notes say the virtual keyboards and mice have not been driven against Windows or a game, so typing, lock-light reports and removal timing are unverified on a live system.
- A virtual keyboard presses the key that types the row's character in the foreground window's input layout, as Windows input does.
- Set both halves back to Windows Input and remove the trial slot.

Sources: Keyboard and mouse models / Virtual keyboards and mice / Keyboard and mouse drawings / Keyboard + Mouse output

20 Check accessibility and translations

Since 5.1.0, PadForge names every control the Tab key reaches for a screen reader and comes in 109 languages. Check what is spoken and what the keys reach, then repeat a page in another language.

Equipment and preconditions

- An installed copy of NVDA and a keyboard. A second language the reviewer reads is optional.
- Use the installed NVDA. PadForge runs as administrator, and a portable copy of NVDA cannot read it unless that copy also runs as administrator.

Procedure

1. Start NVDA, then PadForge. Tab through the Dashboard and record what NVDA speaks for a slot's type tiles, its power button and the engine button.
2. Open a dialog from the keyboard, such as Add Controller, move through it with the arrow keys and Tab, and close it with Escape. Check that the focus goes back to what opened it.
3. On a controller's page, use Ctrl+Tab to change tabs. On a sensitivity curve, move a point with the arrow keys and Page Up and Page Down, and record the spoken point.

4. Start or stop the engine and record whether NVDA speaks the status change while PadForge is the active window.
5. Pick another language at the top of Settings. Check the same page for English left where a translation has no string and, in Arabic, Central Kurdish, Persian or Urdu, for the right-to-left layout with the controller drawings still left to right.

Result and evidence

A passing result records the spoken names and states, the keys that reached each control, and the language checked. A screenshot of a translated page shows the strings, not their quality.

- Record the NVDA version, the PadForge build, the pages and controls checked, and the language.

Limits and reset

- PadForge is tested with NVDA. Other screen readers have not been tried.
- No native speaker has reviewed the translations, and a partial translation shows English where it has no string.
- Set the language back and close NVDA.

Sources: Screen readers and the keyboard / NVDA and administrator rights / What you hear / Keys / Sensitivity curves

21 Map a driving game's telemetry and play a trigger effect from it

Game Telemetry reads what a driving game sends about itself and shows it as a Game Telemetry row on the Devices page. Check that a reading follows the game and that a mapping or macro acts on it.

Equipment and preconditions

- A driving game PadForge reads, a gamepad source, and a DualSense for the trigger branch. The Game Telemetry page lists the games and where each one turns its telemetry on.
- The game's telemetry output on and pointed at this PC and the port PadForge listens on. Assetto Corsa and Assetto Corsa Competizione need no port.

Procedure

1. On the Dashboard, tick Listen for Game Telemetry, pick the game under Game or leave Auto-Detect, set Port, and start the engine. The status line names the port and then the game.
2. Drive, and open the Game Telemetry row on the Devices page. Compare Speed, RPM, the pedal axes, Steering and the gear buttons with the game's own display.
3. On a test slot, map the row's Brake axis to a trigger and its Reverse button to a face button. Check both in a controller tester.
4. Add a macro triggered by a Custom Expression whose variable a reads Speed, such as a > 120, and give it an Adaptive Trigger Effect action on a slot with a DualSense.
5. Drive past that speed and feel the trigger, then slow below it and check that the effect ends.
6. If another tool also needs the game's packets, set Forward to Port to that tool's port and check that both receive the game.

Result and evidence

A passing result shows the row following the game and a mapped output or trigger effect following a reading. A moving row alone does not show that a mapping acted.

- Record the game and its build, the telemetry setting and port, the readings beside the game's display, and the mapped output or trigger response.

Limits and reset

- The Game Telemetry page says no game has sent to this release on a test PC, and that how each trigger effect feels has not been checked against DSX.
- PadForge listens on every network address and adds a Windows Firewall rule for the port the first time it listens.
- Turn Listen for Game Telemetry off, delete the trial macro, and switch the game's telemetry output back off if it was off.

Sources: Turning it on / Setting up each game / The Game Telemetry row / Telemetry in Custom Expression triggers / Adaptive Trigger Effect / Forward to Port / Game Telemetry limits

22 Let a DSX game mod drive a DualSense

Game mods made for DSX send trigger effects and lights to whatever program answers on port 6969. Check that PadForge answers, which controller a mod reaches, and what happens when the game drives the same controller.

Equipment and preconditions

- A game with a DSX mod installed, and a DualSense, DualSense Edge or DualShock 4 assigned to a slot.
- DSX closed, since both programs cannot listen on one port.

Procedure

1. On the Dashboard, tick Accept DSX Game Mods, leave Port at 6969 and start the engine. The line reads Listening on :6969.
2. Start the game. When the mod sends, the line names its program and the virtual controller its effects reach.
3. Trigger an in-game event the mod maps to a trigger effect, feel the trigger, and watch the lightbar and player lights.
4. With PlayStation controllers on two slots, note which one the mod reaches. Its first controller is the first virtual controller in Dashboard order with one online.
5. If the mod checks for DSX before it sends, tick Run a DSX.exe Stand-In and try again.
6. Click Clear Mod Effects on the controller's Devices panel and check that the pad page's own trigger settings return.

Result and evidence

A passing result shows the mod's program on the status line and the matching trigger or light change on a named controller. A listening line alone does not show that a mod sent anything.

- Record the mod and its version, the game, the status line, the controller and slot, and the trigger and light response.

Limits and reset

- The Game Mods page says no mod has yet been run against this release on a real DualSense, and DSX's ten named trigger presets were never published.
- A game's own trigger effects, sent through a PlayStation virtual controller, come before a mod's.
- Turn Accept DSX Game Mods off, which also ends the stand-in. The port files PadForge wrote stay, and DSX writes its own when it runs.

Sources: Turning on game mods / Where mods find the port / Run a DSX.exe Stand-In / Which controller a mod drives / How long a mod's effects last / Game mod limits

23 Feel DualSense haptics on a pad that only rumbles

Some games play a DualSense's haptics as audio and send no rumble. On a DualSense (PS5): Full or DualSense Edge (PS5): Full virtual controller, PadForge turns that audio into rumble for the other controllers on the slot. Check which controllers feel it and which play the haptics themselves.

Equipment and preconditions

- A game that plays DualSense haptics over USB, an Xbox controller or DualShock 4, and optionally a DualSense.
- A slot set to DualSense (PS5): Full with the rumble pad assigned.

Procedure

1. Start the game and confirm that it sees a DualSense.
2. Trigger an event the game plays as haptics. On the pad's Force Feedback tab, watch the RAW meter, which includes haptics played as audio.
3. Feel the pad's left and right motors. Then set its Overall Gain to 0 and check that the haptic rumble stops.
4. If a DualSense is also on the slot, check that it plays the haptics on its own actuators and takes none of the rumble.
5. On the Bass Shakers tab, turn on Route Rumble to an Audio Output and check that Low Motor and High Motor follow the same event.

Result and evidence

A passing result shows the meter and the motors moving with the game's haptic events, and a DualSense on the slot playing them on its actuators. The rumble follows loudness and does not reproduce the haptic waveform.

- Record the game, the virtual preset, the RAW meter during the event, each controller's response, and the Overall Gain used.

Limits and reset

- This path has not run from a game on the bench yet.
- A Steam or Switch controller with Play DualSense Haptics on This Controller on plays the haptics itself and takes none of the rumble.
- Set the slot back to its previous preset and turn off the trial audio routing.

Sources: DualSense haptics played as audio / Live motor activity / Bass Shakers and DualSense haptics

24 Route a DualShock 4's audio by model and link

A DualShock 4's audio depends on its model and how it connects. Check the speaker and headphone routes over Bluetooth, the headset microphone, and the headphone-only path on a cable or Sony's adapter.

Equipment and preconditions

- A DualShock 4 on Bluetooth and wired headphones with a microphone. For the USB branch, a second-model DualShock 4 (CUH-ZCT2) with a cable, or Sony's USB wireless adapter.
- A PlayStation slot with the DualShock 4 assigned.

Procedure

1. On Bluetooth, open the slot's Audio tab and turn on Mirror System Audio. Play a quiet source with Output Path at Default and listen to the speaker.
2. Plug the headphones in. Set Output Path to Headphones (Stereo), then to Follow Headphone Jack, and check the jack and the speaker as you unplug and replug them.
3. With a virtual controller that carries audio on the slot, open Windows' recording devices and speak into the headset microphone. Then register a voice phrase and check that it fires.
4. On a cable with a second-model pad, or on the adapter, play a macro sound with headphones plugged in and again with them unplugged. With nothing plugged in the sound leaves the pad.

Result and evidence

A passing result names the model, the link and the route, with sound heard where the route says and the phrase recognized. The tab's controls alone do not show the route.

- Record the model (CUH-ZCT1 or CUH-ZCT2), the link, the Output Path, what played on the speaker and the jack, and the Windows microphone the virtual controller shows.

Limits and reset

- A first-model DualShock 4 has no audio on a cable.
- Headphone Volume and Output Path are Bluetooth rows, and on a DualShock 4 Output Path has no Headphones + Speaker.
- No DualShock 4 has run against these audio paths on the bench yet.
- Turn Mirror System Audio off and remove the trial phrase.

Sources: DualShock 4 audio links / DualShock 4 headset jack / First-model DualShock 4 limit / Bluetooth DualShock 4 microphone

25 Drive a Novint Falcon or a keyboard screen

PadForge 5.2.0 drives two kinds of specialty hardware: the Novint Falcon, a force-feedback controller whose handle moves in three dimensions, and the screens on Logitech's GamePanel devices, the G19, Arx Control and the Mad Catz S.T.R.I.K.E. 7. Use whichever is available.

Equipment and preconditions

- For the Falcon: the Falcon with its power supply, Force Dimension's SDK installed, and the x64 build of PadForge.

- For a screen: the keyboard, plus Logitech Gaming Software 8.52 or later for Arx Control, or Mad Catz's software and Strike7API.dll for the S.T.R.I.K.E. 7.
- A test slot that the device can be assigned to.

Procedure

1. Falcon: in Settings tick Drive Novint Falcon. The card names the library that loaded. Home the handle by pulling it all the way out and pushing it all the way in.
2. Assign the Falcon's row to an Xbox slot, move the handle and watch the row's three axes. Pick the Joystick feel and feel the centering spring.
3. Play a game event with rumble and feel the handle shake. Hold the Kick button and repeat.
4. Screen: in Settings tick Drive Keyboard Screens. Open the screen's row on the Devices page, set it to Clock, and check the screen.
5. Map one of the keys beside the screen to a face button and check the output.

Result and evidence

A passing result names the hardware, the library or path it runs through, and the observed force or picture. A row that appears does not show that the force or the screen works.

- Record the SDK or Logitech Gaming Software version, the Falcon's axes and force, and the screen's picture and a mapped screen key.

Limits and reset

- No Falcon and none of these screens has run against PadForge yet.
- The ARM64 build cannot load Force Dimension's library and cannot drive Arx Control or the S.T.R.I.K.E. 7.
- Turn Drive Novint Falcon and Drive Keyboard Screens off. Turning the screen switch off gives a G19's screen interface back to Windows.

Sources: What the Falcon needs / Turning on the Falcon / How the Falcon feels / A game's force on the Falcon / The screens / Turning on keyboard screens / Each screen's row / The screen keys

Write a result another reviewer can reproduce

Report the narrow observation first: a control registered, a recipient accepted an identity, a physical effect occurred, or a measured interval was observed. Keep the exact configuration beside it. A closed support issue, a maintainer implementation note, or a publisher's physical-controller feature list does not become a fresh test of this setup.

For timing, name the reference and measurement path. The engine's one-millisecond target is not a one-millisecond input-to-game result. A Bluetooth buffer setting is not milliseconds. Speaker-relative observations, round-trip experiments, microphone latency, and haptic delay are different measurements.

Preserve failures that remain after the documented preconditions are satisfied. Record what was visible at the physical-source, PadForge, and receiving-application stages without declaring the cause from one silent view. For a driver-layer claim, a parsed UI alone is not raw reception evidence.

Field	Record
Setup	Build, Windows/architecture, physical model/firmware/transport, output preset, dependency and recipient versions
Action	Exact source, mapping/layer/profile state, recipient event, and release step
Observation	Configured only, observed in PadForge, observed in recipient, observed physically, or measured by the stated method
Result	Pass, partial, failed, skipped, or unestablished, with the precise affected capability
Return path	How the original profile, bindings, hiding state and vendor settings were restored

- Keep raw private evidence separate from a publishable result. Remove serials, device paths, personal executable paths, network addresses, pairing/access codes, and tag UIDs from shared material.
- Do not treat 1,110 numeric identity pairs, 465 mapping names, 337 bundled profiles, 136 selectable controller profiles, 51 keyboard and 52 mouse models, or 96 Extended profiles as tested retail-device totals.
- A release-specific conclusion should name the build and date. Record a later retest separately instead of rewriting a historical report as if it used the new version.

Sources: Polling rate / Supported Devices / PadForge 5.0.0 release notes / Extended (HIDMaestro)