

MIDIDeviceManager

What's New

Version 2.3 — release notes for Build 111, Build 108, Build 107, Build 106 and Build 105.

This document covers the five most recent releases, newest first. The Build 106 and Build 105 notes follow, kept as they were written.

Build 111

Build 111 puts MIDIDeviceManager on your wrist, and fixes one thing that could quietly cost you a sort order.

MDM arrives on the Apple Watch

MIDIDeviceManager now runs on the Apple Watch as a full companion app — Devices, Scenes and Live, staying in step with your phone or iPad in both directions. Tap a Pad on either device and it highlights on both, the same full colour change you'd see tapping it directly, not a dimmed or brightened version of the Pad's own colour. There's no six-Pad limit on the Watch; scroll for as many as a Device has.

Because the Watch and phone sync independently rather than simply mirroring one another, they can be pointed at two different pieces of equipment at once — one hand on the phone, the other glancing at the wrist, without juggling a single screen between them. The Watch's Devices page follows whichever Device sits first in your phone's Devices list; reorder that list and the Watch follows.

Two things worth knowing: the very first tap of a session, after a period of inactivity, doesn't always register on the Watch immediately, though every tap after that one works reliably. And if the Watch shows "iPhone unreachable," this is almost always just the Watch's own screen having gone to sleep — tap it to wake, and the connection returns.

Undo Last Sort now remembers what you asked it to

Undo Last Sort on a Device's Pads screen used to live in temporary screen state — lost the moment you left the screen, backgrounded MIDIDeviceManager, or tapped Sort A–Z a second time before undoing, which silently overwrote the true original order with the already-alphabetised one. It's now persisted properly, keyed to the Device, and survives all of that.

Build 108

Build 108 gives you two new ways to shape a sound mid-performance, and is otherwise about being told rather than left to notice: MIDIDeviceManager now names a Device that drops out of a background session instead of staying quiet about it, makes it harder for a Device to report itself

present when it isn't, and opens the controls that used to require Advanced or Pro mode to nearly everyone from the first launch.

New controls beyond the Pad

Devices can now carry sliders and on/off buttons of their own, separate from Pads and available from Edit Device. They stay in effect after a Pad recalls a patch — set with a Pad, then adjusted afterwards with a button, the same way you'd reach for a pedal mid-song. One thing worth knowing: a button always does the same thing when you tap it, because MIDIDeviceManager can't see what your instrument is actually set to — on and off are two independent sends, not one control that remembers where it left things. Scenes show the same controls automatically, for whichever Devices that Scene uses, once it's been triggered.

A Pad can now carry a Control Change as well

Also Send a Control Change, in Edit Pad, lets a Pad send one fixed Control Change alongside its Program Change — setting a level or turning an effect on at the same moment you recall the patch.

Worth knowing if you haven't tried it: a Pad can drive another app

MIDIDeviceManager can control any other app on the same device that accepts MIDI — Stage Traxx, BS-16i, AmpliTube among them. If you play through an app as well as hardware, a Pad reaches it the same way it reaches an instrument.

A dropped Device now tells you, even in the background

While a background session is running, MIDIDeviceManager used to rely on you noticing a Device had gone quiet. It now names the Device directly on the Dynamic Island the moment it drops, and sends one silent notification the first time that happens. Nothing to turn on — it works automatically whenever a background session is active.

Red now means what it always promised

A Device with no way to reach your equipment — its assigned port gone, nothing to fall back on — was showing the same orange as a Device quietly using a fallback port and still sending. Individual Devices can now show red, not only the badge that summarizes them, so orange and red mean what MIDIDeviceManager has always said they mean: orange is still working, red is not.

A Device that goes quiet now finds its way back

A Device that stopped answering entirely — genuinely disconnected, no replies at all — used to exhaust a fast automatic retry and then stay orange indefinitely, no matter how long you waited. The only way back was a manual Reset MIDI Connections. MIDIDeviceManager now keeps trying on a slower cadence, tied to your own Auto Check setting, so a Device that comes back usually confirms itself without you having to do anything.

A Device can no longer borrow another one's identity, unless you say so

On a shared connection, if exactly one reply came back and it didn't match the Device MIDIDeviceManager expected, it was sometimes accepted anyway — so a Device already known to MIDIDeviceManager could show green from an unrelated interface's own reply, even while the real instrument was switched off. This is fixed for any Device with a confirmed history; only a genuinely new Device still gets the benefit of the doubt on a single reply. Trust This Connection, introduced in Build 105, still lets you accept that risk deliberately on a simple, single-instrument connection — it's a choice available to you, not a gap that's been closed everywhere.

Triggers arrive for everyone, and now light up what they fire

Live Pad MIDI Triggers were behind Advanced mode until this build — for most testers, this is the first release where a footswitch can fire a Pad at all. And when it does, the Pad or Scene it fires now highlights the same way a manual tap does; before this build, only tapping by hand lit anything up.

Nearly everything is available from the first launch

Easy, Advanced and Pro are gone. MIDIDeviceManager now has two modes, Standard and Pro, and most of what used to require Advanced is available from the start — triggers, Also Send a Control Change, the MIDI Trigger Manager, MIDI Health Monitor, Network MIDI. Pro now means one thing: the SysEx Lab, for building and sending raw SysEx by hand.

Guided device import, open to everyone

Building a Searchable Instrument Index directly from an instrument used to require Advanced or Pro mode. It's now reachable from Standard too, through a simpler screen that shows only what that import needs and returns you to Libraries once it's done.

One searchable list, for everyone

Choosing a supported instrument to import from used to mean two different pickers depending on your mode. There's now one manufacturer-grouped, searchable list, shared by both — and an instrument can now appear under more than one manufacturer where that's genuinely true.

A Pad can carry its own colour

Edit Pad now has a colour picker of its own, separate from the colour a Pad gets automatically. Pick whatever helps you find it fastest on stage; one tap returns it to where it started.

Smaller things

- Switching a Device off now keeps its assigned ports. Switch it back on and it reconnects to where it was, rather than asking you to reassign them.
- If MIDIDeviceManager notices two manufacturer names in your list look like the same company, it now offers to group how they're displayed — your Devices themselves are never changed.
- In MIDI Health, Reconnect BLE MIDI now precedes Refresh Device MIDI Connections.

- Several buttons — including Reconnect BLE MIDI and Copy Diagnostics — briefly confirm themselves with a checkmark and a changed label before returning to normal, so a tap no longer leaves you wondering whether it registered.
- A Device now recognises its own port by name even if iOS assigns it a different internal ID — after a reconnect, or a system update — and reconnects on its own rather than showing as missing.

Build 107

Build 107 is about the first ten seconds. MIDIDeviceManager now waits for your instruments rather than for a clock, tells you which connection is missing when one is, and opens its editors without the pause that used to come first.

The app arrives with your studio ready

The opening screen used to run for a fixed fifteen seconds whether or not your instruments had answered. It now waits for them, shows each one as it confirms, and closes as soon as they have settled — sooner on a simple rig, longer on a large one.

An instrument that does not answer no longer holds anything up. Some instruments do not respond to an identity check at all, and MIDIDeviceManager no longer waits for them at startup — it says so on the Devices screen and opens. It keeps checking quietly afterwards, so an instrument that answers late will turn green on its own.

And it names what is missing. MIDIDeviceManager records the name of every MIDI port it has seen, so when an adaptor is disconnected it can tell you which one rather than only that something is absent. This works from the second launch after updating onward — on the first there is no record to compare against yet.

Devices confirm on the first attempt

A Device is checked against a table of manufacturers when it answers an identity request. On some rigs that table had not finished loading when the first check ran, so the check failed, the Device went into a recovery cycle, and it confirmed a cycle later on a reply identical to the one already received. Nothing was wrong with the instrument; only with when we asked.

Whether you saw this depended on how the timing fell on your device, so some testers will have noticed nothing. Where it happened, it happened on every launch. Devices now confirm on the first attempt.

Editors open immediately

The first long-press of each session used to take about a second and a half before the editor appeared, and every one after it was instant. The preparation that happens behind the opening screen now covers that too, so the first is as quick as the rest.

Identity checks every thirty seconds

MIDIDeviceManager periodically asks each instrument whether it is still answering. That question was being asked every five seconds, which on a five-instrument rig meant it never stopped — and on Bluetooth that traffic competes with the notes you are playing. The interval is now thirty seconds, which still notices a dropped instrument well inside a song.

If you are already testing MIDIDeviceManager, this change will not reach you automatically. Your existing setting stays as it is. To take the new interval, open Tools, then Settings, and under MIDI Device Confirmation set Auto Check to Every 30 seconds. That setting is now visible in every user mode; it used to be hidden unless advanced live routing was switched on.

Smaller things

- Editing a Device no longer rewrites its patch list when nothing in that list has changed.
- Saving happens in the background rather than between keystrokes, so typing in a name field no longer waits for the save to finish.
- Choosing Manual for Auto Check now says what it means: MIDIDeviceManager will not notice on its own if an instrument stops answering.

Build 106

Build 106 is about what happens when MIDIDeviceManager is not the only thing you are looking at, and about a scan that was quietly getting it wrong.

MIDI can now keep running on iPhone

Build 105 told you that on iPhone MIDIDeviceManager had to stay on screen. On iOS 26 and later, that is no longer true.

Open Keep MIDI Running in Settings and tap Start Background Session. MIDIDeviceManager then keeps receiving MIDI while another app is in front — lyrics, chord charts, backing tracks. Start it before you play rather than during a song.

While a session is running, a BG badge appears beside the MIDI health indicator. Green means MIDI is protected and the session is running. Amber means the session is starting or restarting — normally brief, and never means the session has ended. No badge means no session, which is the normal state and not a fault.

Red is the one to know, and you should rarely see it. A session you stop yourself never shows red — stopping from Keep MIDI Running inside the app clears the badge cleanly, and stopping from the Dynamic Island closes MIDIDeviceManager entirely, so there is nothing left running to show a badge at all. Red means the system itself is struggling to keep a session running that you never told it to stop, even after MIDIDeviceManager tried to bring it back automatically.

A long session does not need to be babysat. MIDIDeviceManager does not treat a fixed length as a cutoff, and if the system ever does need to end a session, it tries to bring it back on its own first. A lock-screen prompt may also appear after about two hours; accepting it keeps the session alive. If

you would rather not think about any of this during a long service or a long set, starting a fresh session during a break costs nothing and is always safe — just not something you need to do.

iOS shows the session in the Dynamic Island, from any app. Swiping the indicator away only hides it; the session keeps running. Stopping the session from the indicator ends it and iOS then closes `MIDIDeviceManager`, so you return to a cold start with MIDI and Bluetooth reconnecting — avoid this between songs. Stopping it from Keep MIDI Running inside `MIDIDeviceManager` ends the session and leaves the app running, and that is the one to use. Tapping the indicator expands it, and tapping again takes you straight back to `MIDIDeviceManager`.

On iPad, nothing has changed and nothing needed to: keep `MIDIDeviceManager` on screen beside the other app in Split View and it is never suspended. On iPhone below iOS 26 there is still no background session, and `MIDIDeviceManager` must stay on screen.

The MIDI Studio scan now creates every device you selected

Asked to create six devices from a scan, `MIDIDeviceManager` could create three and stop without reporting anything. Some of the devices it did create carried identities belonging to other instruments.

This is fixed, and it is worth saying what it was, because none of it was what we expected. Three separate faults were found, and five theories were argued and discarded before the code was instrumented.

Instruments sharing one interface were being judged duplicates of each other, so the second was discarded. A device's listening port was being guessed from the name of its sending port, so every instrument behind one interface reported whichever one owned that port. And identity checking ran at the same time as the scan, so a reply arriving mid-scan was recorded as though the scan had produced it. A first scan on a clean install was unaffected, which is why this went unseen for so long.

Separately, an instrument reachable by several routes was offered once per route — a Line 6 POD 2 appeared five times. It is now offered once.

A scan that finds nothing now says so, instead of returning in silence.

MIDI channels are settled before a device is created

Two devices on the same port and the same MIDI channel will both answer the same message, and the result looks like an instrument misbehaving rather than a setting being wrong. This is the fault that looks least like a fault.

`MIDIDeviceManager` now asks you to resolve MIDI channels before creating devices, and will not proceed while a collision remains. If you are deliberately sharing a channel — two sound modules you always want to answer together — confirm it and `MIDIDeviceManager` will accept it.

Building an index has a clearer starting point

The screen that offers the ways of building a Searchable Instrument Index is now Create Searchable Instrument Index, and it names the four routes plainly: Manufacturer Resource, Supported Device, CSV / Excel, and Patch Walk MIDI Mapping. Each says what it does and where it comes from.

Supported Device — reading an instrument directly over MIDI — appears in Pro Mode. The other three do not require it.

One thing worth knowing before you start. An index import asks your instrument for its whole patch list, several hundred replies one after another. Bluetooth MIDI can drop replies during a transfer of that length, and every dropped reply becomes a patch missing from the finished index — while the import still appears to have worked. Use a wired connection when you build an index. Bluetooth is fine for playing afterwards.

Smaller things

- A patch stepper in Edit Pad. Arrows walk the filtered patch list and send each patch as you go, so you can hear what you are choosing.
- Trust can now be withdrawn. Build 105 let you trust a MIDI interface and never take it back. Edit Device now lets you undo that decision.
- Where sending and listening use different hardware, both are now shown, on two lines, rather than one standing in for the other.
- The Line 6 POD 2 factory library now appears only when a POD 2 device exists, and arrives locked.
- Copy Diagnostics in Settings, for when you are writing to support.
- In the last week of the trial, the countdown now also appears inside the application above the tab bar — not only on the startup screen, which a musician who never quits MIDIDeviceManager would never see.

Build 105

Build 105 does two things. It improves how MIDIDeviceManager behaves with real-world MIDI equipment, routing interfaces and Bluetooth adaptors, refining workflows identified during physical hardware testing and TestFlight validation.

And it makes Searchable Instrument Indexes real. Three instruments can now be indexed — the ASM Hydrasynth, the Line 6 Helix and the Korg minilogue xd — where before there were none. Two of those are built from your own instrument.

Every change described here was validated on physical hardware, and several were confirmed independently by TestFlight testers using their own instruments.

MIDI interfaces are now recognised as devices

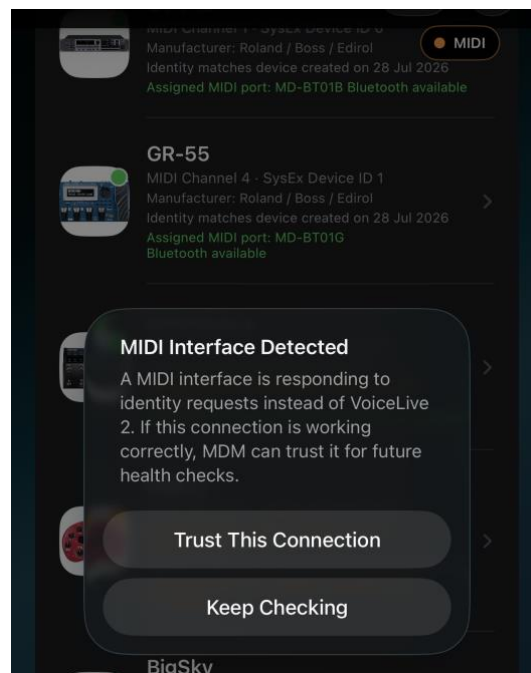
If you connect your instruments through a MIDI interface — an EDIROL UM-880, a CME WIDI Jack, or similar — you may have seen MIDIDeviceManager report a device mismatch even though everything was working perfectly.

The reason turned out to be simple. When MIDIDeviceManager asks a connected device to identify itself, some interfaces answer on their own behalf rather than passing the question through to your

instrument. The UM-880, for instance, identifies itself as a Roland device. MIDIDeviceManager was comparing that reply against the instrument it expected to find, and reporting a problem that did not exist.

Build 105 introduces the Trusted MIDI Interface model. MIDIDeviceManager now understands the difference between an instrument's identity and an interface's identity.

When it recognises an interface answering on its own behalf, MIDIDeviceManager marks that device with an orange MIDI badge instead of the usual green dot, and asks you what to do:



MIDI Interface Detected — shown here for a VoiceLive 2 reached through a multi-port interface

The orange badge is not an error. Nothing has gone wrong, and your instruments continue to play. MIDIDeviceManager is telling you it can see an interface answering identity requests on behalf of your instrument, and asking you to confirm that this is your setup.

Choose Trust This Connection if the connection is working correctly. MIDIDeviceManager will accept that interface's replies from then on and the device returns to green. Choose Keep Checking to leave things as they are — the device stays marked and MIDIDeviceManager keeps testing it, which is the right choice if you are not sure the connection is behaving.

Trusting is normally a one-time operation. MIDIDeviceManager remembers the decision, and it survives quitting and relaunching the app, disconnecting and reconnecting the interface, and powering the interface off and on again.

Most musicians will never encounter this dialog. It appears only when an interface answers identity requests itself, which in practice means a multi-port MIDI interface or certain Bluetooth adaptors. Currently recognised interfaces are the CME WIDI Jack and the Roland / EDIROL UM-880, and the design is deliberately extensible as further interfaces are validated on hardware.

Searchable Instrument Indexes, built from your own instrument

A Searchable Instrument Index lets you find a sound by name instead of remembering where it lives. Build 105 is the release where that became genuinely useful, because MIDIDeviceManager can now build one from your instrument rather than from a generic list.

That distinction matters. An index built from your own instrument contains your patches, under the names you gave them, in the slots you put them in. Rename a sound or reorganise a bank, build the index again, and it stays correct. A published factory list can never do either.

ASM Hydrasynth

Save a SysEx dump from the instrument and import the .syx file. Your own patches are indexed alongside the factory ones, with patch names and MIDI addressing preserved, ready to use throughout Devices, Pads, Scenes and Live Mode.

One note on how this arrived. During testing, a bank numbering error meant that selecting a patch in bank A recalled the equivalent patch in bank B. Tester Eric found it on his own instrument, it was traced to a single arithmetic conversion, and it was corrected throughout — in the parser, the exporter, the reference implementation and the verification material. The correction was then confirmed in both directions on physical hardware. No affected build was ever distributed, and selecting A001 now recalls A001.

Line 6 Helix

The Helix has no SysEx, so the route is the backup file. Create a .hxb backup in HX Edit and import that. Your own setlists are indexed alongside the factory ones.

Korg minilogue xd — a factory patch list

The minilogue xd is supported differently, and it is worth being clear about how. Rather than reading a file from your instrument, MIDIDeviceManager imports a CSV list of the factory patches, which you can download from the resource centre at mididevicemanager.com. Look under CSV Device Indexes and choose Korg. The CSV Device Index Import Quick Start Guide, on the same page, walks through the import. The list was compiled from patch data sent to us and checked against the manufacturer's own information.

That gets you searching and recalling factory sounds without typing anything in. What it cannot do is follow your own edits: rename a patch on the instrument or write your own into a slot, and the list will no longer match. For that, an index built from the instrument itself is the answer — which is where the Hydrasynth and Helix routes above lead, and where we would like to take more instruments.

MIDIDeviceManager in the background

Superseded by Build 106. This section describes Build 105 and is kept as it was written. On iOS 26 and later, MIDIDeviceManager can now keep receiving MIDI on iPhone through a Background Session — see MIDI can now keep running on iPhone, above. The rest of this section still holds for earlier system versions.

Testing established clearly how MIDIDeviceManager behaves when it is not the app you are looking at.

- In Split View, alongside another app, MIDI continues to work.
- When MIDIDeviceManager is fully in the background, MIDI reception is suspended.

This is determined by iOS rather than by MIDIDeviceManager — the system suspends the app, and no setting within MIDIDeviceManager can override it.

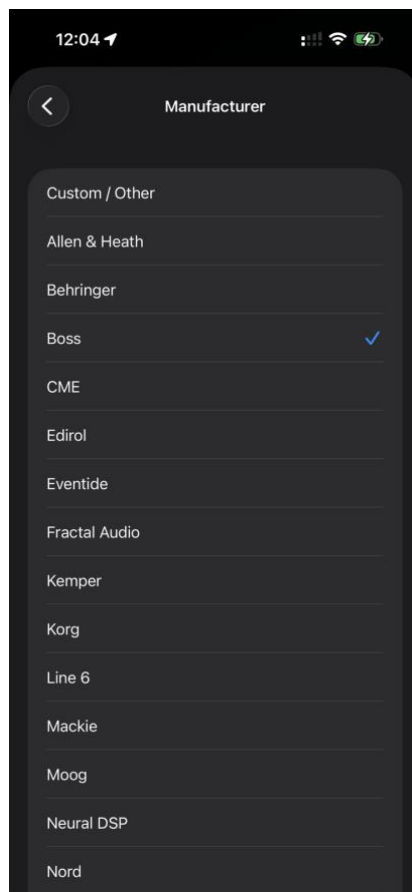
On iPhone there is no Split View. MIDIDeviceManager must stay on screen to receive MIDI. If you need another app in front of you during a song, use an iPad and keep MIDIDeviceManager alongside it in Split View. This is worth knowing before a performance rather than during one.

Apple has been contacted for formal guidance on the supported architecture for MIDI applications of this kind.

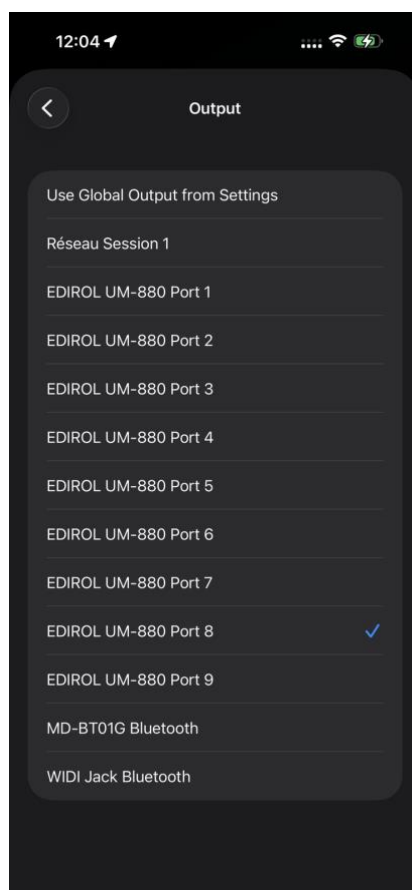
Long menus stay where you put them

Choosing from a long list — a manufacturer, or a MIDI port — could send the menu jumping back to the top while you were scrolling, which made anything near the end of the list effectively unreachable. That is fixed.

Long lists — Manufacturer, and the Input and Output port lists — now open as their own screen rather than as a drop-down menu. Your current choice carries a blue tick. Tap an item and MIDIDeviceManager returns you to where you were. These screens therefore look a little different from Build 104.



Manufacturer



Output — a deliberately large rig: a nine-port interface, two Bluetooth adaptors and a network session to a Mac

The Output list above belongs to an unusually complex setup, kept that way on purpose for testing. Most musicians will see far fewer entries. MIDIDeviceManager handles both.

Smoother throughout

MIDIDeviceManager now redraws its interface only when something has actually changed, rather than continuously while MIDI is flowing. Previously every incoming message and every routine identity check could prompt a redraw, whether or not anything on screen was different.

The result is an app that feels more responsive everywhere and is easier on battery — most noticeably when several devices are connected and traffic is heavy.

Strymon BigSky

BigSky Searchable Instrument Index creation is now fully available over wired MIDI. You can go directly from Create Searchable Instrument Index to Build Complete BigSky Instrument Index, and the import has been confirmed working on physical hardware.

If you have tried a BigSky import previously and found it unavailable, the import engine itself was working throughout — the problem was in how the interface led you to it. Import over Bluetooth remains under investigation; use a wired connection for now.

Multiple CME WIDI devices

CME have confirmed the recommended configuration for using more than one WIDI device with an iPhone or iPad: set every WIDI device to Force Peripheral mode.

In this mode the WIDI devices wait for your iPhone or iPad to establish the connection instead of pairing with each other first. This removes any dependence on the order in which you power devices on, and allows your device to reconnect to them automatically.

This is CME's own guidance rather than a workaround, and it is now the recommended configuration in the CME WIDI Quick Start Guide, available from the resource centre.

This recommendation applies specifically to CME WIDI devices, and should not be assumed to apply to Bluetooth MIDI products from other manufacturers.

More in-app help

The Help Card system introduced in Build 104 has been extended. New cards cover Live Getting Started, Live Setlists and Create Searchable Instrument Index, and the wording of several existing cards has been refined.

Clearer wording for SysEx Definitions

SysEx Definitions and Patch Libraries are two different things, and the interface did not always make that obvious. A SysEx Definition describes how MIDIDeviceManager talks to a particular instrument. A Patch Library holds the sounds you can browse, search and recall.

The wording throughout the SysEx Definition screens has been clarified, and internal filenames are no longer shown where they served only to confuse.

Thank you

Thank you to everyone who tested Build 105. Several of the improvements in this release came directly from musicians using MIDIDeviceManager with real equipment, in real studios and real live-performance environments.

If anything in this build behaves unexpectedly, please write to **support@mididevicemanager.com**.