



MIDI Device Manager

User Manual

Version 2.6 for Build 111

Spend less time managing MIDI. More time making music.

Contents

Contents	2
Part One	5
Why this part comes first	6
What MDM is not	6
1. What is actually connected	6
2. Sending and listening are two different things	7
3. Devices and interfaces are not the same, and MIDI does not always agree	8
4. Why a scan finds some equipment and not all	9
Not everything that answers should be added	10
5. What “works with MDM” actually means	10
What MDM does not do	11
6. A word about the word “index”	12
7. Status colours, and what they are telling you	12
8. When an interface answers for your equipment	13
9. Why MDM cannot tell you everything	14
10. What MDM does before it hands you the application	15
11. What to take from this part	15
Part Two	17
Before you start	18
1. Choosing how your equipment gets in	18
2. When the scan finds nothing	19
3. Adding a Device by hand	19
4. Reading a Device at a glance	20
5. When a Device is green and nothing happens	21
6. What to take from this part	22
Part Three	24
Why bother	25
1. Four ways in, and which one is yours	25
2. All four routes live on one screen	26
3. What an index is worth once you have one	27
4. Making the index find what you meant	27
5. Knowing which patch you are on	28
6. What an index does not do	29
7. What to take from this part	29
Part Four	30
What this part is for	31
1. Three layers, and choosing between them	31

2. Pads worth having	32
Organising a Pad grid	32
3. Sliders and Buttons	33
4. Scenes.....	34
5. Setlists	34
6. Live Mode.....	35
7. Playing without touching the screen	35
Back up what you have built	35
8. What to take from this part	36
Part Five.....	37
What this part is for	38
1. MIDI while you are looking at something else.....	38
2. A second surface: the Apple Watch companion.....	39
3. The badge, and the trap in it.....	40
4. The indicator in the Dynamic Island.....	41
5. When something stops answering.....	41
6. Ten minutes before you play	42
7. What to take from this part	43
Part Six.....	45
How to use this part.....	46
1. MIDI channels, and the number that is not a channel.....	46
2. Program Change	47
3. Bank Select	47
4. Control Change.....	48
5. Identity Request	49
6. System Exclusive.....	49
7. Glossary	50
Appendix A — Where to look when something is wrong.....	52
Nothing is reaching my equipment.....	52
MDM cannot find my equipment	52
Patches and indexes.....	53
While you are playing.....	53
Still stuck	53
Appendix B — Ten minutes before you play	55
1. Find out how much is wrong.....	55
2. Bring MDM to the front	55
3. Tap the MIDI Health badge	55
4. Read the status line, not the colour.....	55
5. Power-cycle the adaptor, not the equipment	56

6. Check the channel on the one that is wrong	56
7. Play without it	56
What not to do with ten minutes to go	56
Afterwards.....	56

If the contents list is blank or out of date, select it and press F9 to rebuild it.

Part One

How MDM Sees Your MIDI World

What MDM can see, what it cannot, and why.

Why this part comes first

Most of what puzzles musicians about MIDIDeviceManager is not really about MIDIDeviceManager. It is about MIDI — a standard designed in 1983, still working, and still carrying the assumptions of the equipment it was designed for.

MDM sits on top of that standard and tries to make it behave like a modern application. Most of the time it succeeds. When it does not, the reason is almost always something MIDI does that no amount of software can hide.

This part explains what MDM can see, what it cannot, and why. Read it once and most of the rest of this Manual becomes obvious. Skip it and you will eventually meet something that looks like a fault and is not.

This part contains no procedures. Where a task is named, the Quick Start Guide that carries the steps is named with it.

Your trial begins the moment you create your first Device — not when you install the app.

What MDM is not

MDM is not a patch editor. It does not build sounds or adjust the parameters inside them. The software your manufacturer wrote for exactly that purpose remains the right tool for making a sound in the first place — MDM begins its work once that sound already exists.

MDM is not a manufacturer librarian. It does not back up, restore, or manage the contents living inside your equipment the way dedicated librarian software does. What MDM keeps of its own — Devices, Pads, Scenes, Setlists, indexes — is a separate thing entirely: its own record, not a copy of what your equipment holds.

MDM knows what the sound is called and where to find it. The instrument knows how the sound is built.

1. What is actually connected

MDM does not see your equipment. It sees ports.

A word on what this Manual calls things. Equipment means anything of yours that speaks MIDI — a synthesizer, an effects pedal, a rack unit, a drum machine, a lighting controller, another app on the same iPad. MDM does not care which, and neither does this Manual. **Device**, with a capital D, means something different and more specific: the entry you create inside MDM to represent a piece of that equipment. You will see both words often, and the difference between them is the subject of this part.

A port is a route in or out of your iPhone or iPad through which MIDI can travel. Your equipment might be at the far end of one, or it might not — MDM has no way of knowing until something answers.

There are four ways a port comes to exist.

A direct USB cable. Your equipment plugs straight into the iPad or iPhone and appears as a port of its own. This is the simplest arrangement and the one with the fewest things to go wrong. It requires the

equipment to be class-compliant: able to describe itself to the operating system without a driver. A great deal of equipment is. Some, including units that are otherwise perfectly modern, are not — and when they are not, no setting on either side will make the USB socket work. Their 5-pin MIDI sockets still work perfectly.

A MIDI interface. A box with 5-pin MIDI sockets on one side and USB on the other. Your equipment plug into it and it plugs into your iPad. A single-port interface gives you one port; an eight-port interface gives you eight, and MDM sees all of them. This is how most equipment made before USB reaches an iPad or iPhone, and how many pieces made since still do.

A Bluetooth adaptor. A small adaptor that plugs into your equipment's 5-pin MIDI sockets and connects to your iPad wirelessly. From MDM's point of view the result is another port. Bluetooth introduces two things a cable does not: a connection that can drop, and a limit on how much data can pass reliably at once. Both matter, and both are covered later.

A network session. MIDI over Wi-Fi to a computer on the same network. Less common in performance and mentioned here for completeness.

What matters is that all four produce the same thing: a port. Once a port exists, MDM does not care how it got there. Equipment on a Bluetooth adaptor and equipment on a USB cable are handled identically. This is why advice about one usually applies to the other.

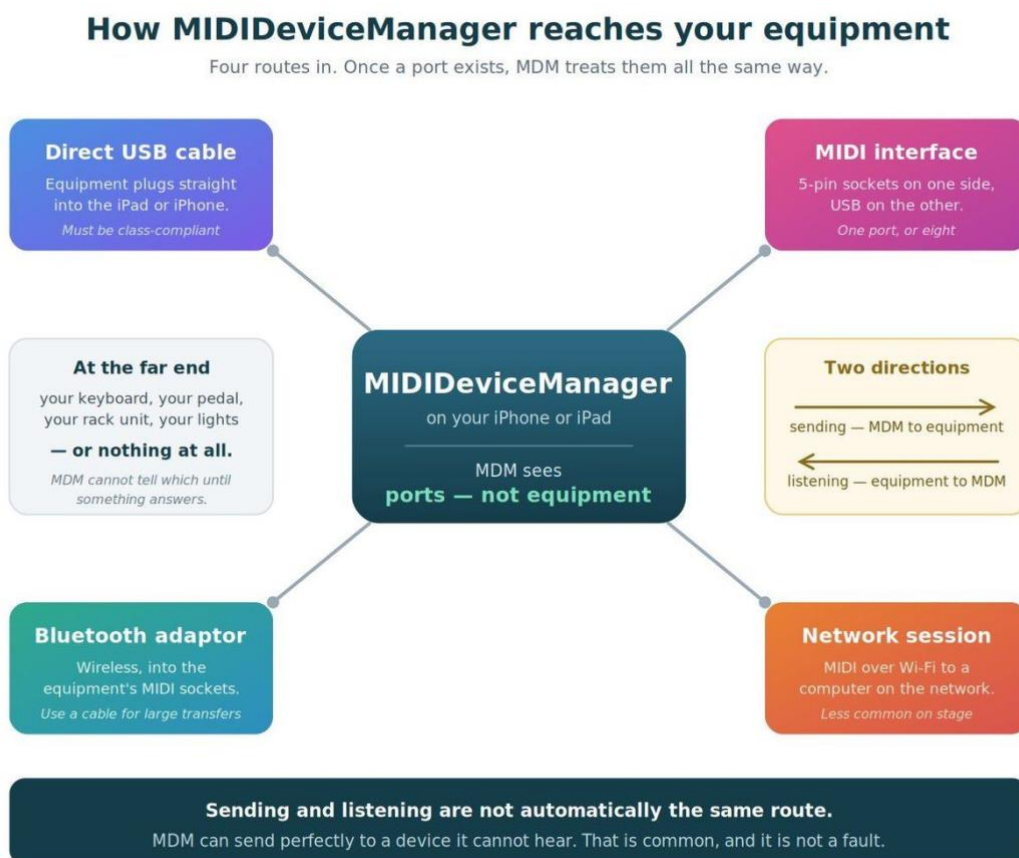


Figure 1.1 — Four routes in, and one thing MDM actually sees.

2. Sending and listening are two different things

This is the idea that unlocks the most confusion, and it is worth reading twice.

Every MIDI connection has two directions, and they are separate. **Sending** is MDM telling your equipment to do something — change to this patch, move this control. **Listening** is your equipment telling MDM something — this is who I am, this is the patch I am on now.

They are not automatically the same route.

A Device in MDM stores both: which port it sends on, and which port it listens on. On a simple setup they are the same piece of hardware and you will never think about it. On a larger one they may not be.

Why this matters in practice. MDM can send perfectly to a device it cannot hear. Program Changes will work, patches will recall, and everything will look correct — while MDM has no idea what the equipment is or what patch it is currently on. That combination is common and it is not a fault. It simply means the return path is not connected, or the equipment does not answer.

The reverse is rarer, but possible.

Where sending and listening use different hardware, **MDM shows both**, on two lines, so you can see what is actually configured rather than assuming.

3. Devices and interfaces are not the same, and MIDI does not always agree

Here is where MIDI's age shows.

When MDM wants to know what is at the end of a port, it sends a question called an **Identity Request**. It is part of the MIDI standard, it has been there since the beginning, and a device that supports it replies with its manufacturer, its family and its model.

Two things complicate this.

First: not every device answers. Answering an Identity Request is optional, and always has been. A great many devices — including current, professional, expensive ones — simply do not implement it. They receive Program Change perfectly, they change patches instantly, and they will never tell you their name.

Second: sometimes the wrong thing answers. A multi-port MIDI interface is itself a MIDI device, and some interfaces answer an Identity Request on their own behalf rather than passing the question through to the equipment behind them. MDM asks who is there, and the interface says I am — quite truthfully — while your equipment sits silently behind it.

Neither of these is a fault. Both are MIDI working exactly as specified. But both mean the same thing for you: **what MDM can discover about your rig is not the same as what your rig can do.**



Figure 1.2 — One question, three possible answers, and only one of them a problem.

4. Why a scan finds some equipment and not all

Scan MIDI Studio asks every port in turn who is there, and offers to create a Device for everything that answers. The Quick Start Guide carries the steps.

It is the quickest way to set up when it works. It is also, for the reasons above, incomplete by nature. It can only find devices that answer an Identity Request, and it can only identify them correctly when the reply came from the equipment rather than from something in between.

So a scan produces one of four outcomes.

- **It answers, and answers as itself.** MDM creates a Device with the right manufacturer and model, already knowing what it is. Nothing more to do.
- **It answers, but the reply came from an interface.** Scan does not try to tell the two apart. An interface that answers for itself is listed exactly like any other candidate device — nothing marks it, and nothing warns you at this stage. Recognising it and leaving it switched off is yours to do; see below.
- **Something answers that is not the equipment you meant to add.** See below.
- **It does not answer.** The scan passes over it. Nothing is created, and nothing is reported, because from MDM's point of view nothing happened.

Scan MIDI Studio not finding a device does not mean the

device is incompatible with MIDIDeviceManager.

This is the most important sentence in this Manual and it deserves the emphasis. A device that never answers a single Identity Request can be added by hand in under a minute and will then work exactly as well as one that was discovered automatically. Pads, Scenes, Setlists, Live Mode, Searchable Instrument Indexes — all of it. The only thing it cannot do is introduce itself.

Adding a Device by hand is not a workaround. It is one of two equally normal ways to set MDM up, and for a good proportion of equipment it is the only way. The [Quick Start Guide](#) carries the steps for that too.

Not everything that answers should be added

A scan reports what answered, not what an answer is. A multi-port interface or a Bluetooth adaptor that replies with its own identity appears in the results looking much like the equipment behind it. An interface is itself a MIDI device, and a perfectly legitimate one — something answered, on a port, with a manufacturer and a model. Nothing in MIDI distinguishes the box carrying your equipment's messages from the equipment itself, so MDM cannot make that judgement for you.

Add the equipment. Leave the interfaces and adaptors switched off. The name in the reply is usually the clue — a row calling itself an interface generally is one. Your adaptors do not need Devices of their own: they are the road, not the destination.

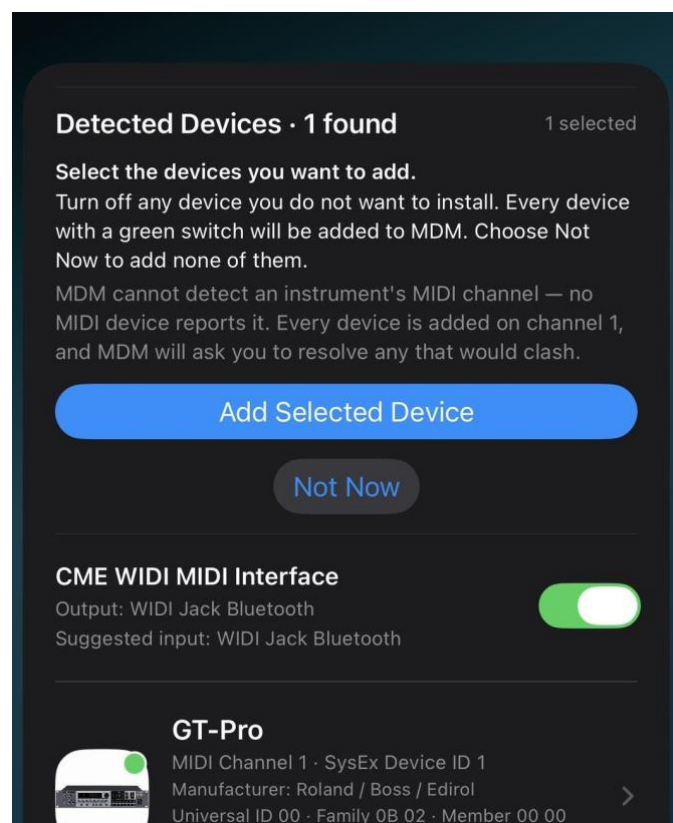


Figure 1.3 — A scan result. The interface has answered for itself and is offered alongside the equipment; leave it switched off.

5. What “works with MDM” actually means

There is no single answer, because there is no single thing a piece of equipment has to support. Yours may work with MDM in any of these ways, and most work in several.

- **It responds to Program Change and Control Change.** This is the floor, and almost everything with a MIDI socket clears it. If your equipment can change patch when a foot controller tells it to, it can change patch when MDM tells it to.
- **It connects by DIN but not by USB.** Common in devices that predate USB, and in some that do not. Use a MIDI interface and everything works.
- **It connects by class-compliant USB.** Straight into the iPad, no interface needed.
- **It connects by Bluetooth through an adaptor.** Any device with 5-pin sockets can, with the right adaptor.
- **It answers Universal Identity Requests.** Convenient. A scan finds it and names it.
- **It works correctly without answering Identity Requests.** Equally functional; add it by hand.
- **It has a route to a Searchable Instrument Index.** Its patch names can be brought into MDM so you can recall patches by name rather than by number.
- **It supports deeper direct-from-device operations.** Reading its patch list over MIDI, and other advanced work.
- **It is not hardware at all.** MDM does not treat an app's own MIDI port any differently from a piece of equipment's. If another app on your iPhone or iPad implements MIDI, MDM sees its port the same way and can communicate with it — to do whatever that app itself is designed to do with MIDI. What that turns out to be is up to the app, not MDM.

Reading down that list, notice that the first item is enough. Everything below it is convenience, and the convenience varies by device. **The question is not whether your equipment is supported. It is which of these things it can do, and what that means for how you set it up.**

What is known about a particular piece of equipment — including manufacturer settings that must be changed before anything works at all — comes from hardware we have tested ourselves and from what musicians report back. It is not a complete picture and it never will be: there is more MIDI equipment in the world than any one project can test. If you are unsure about yours, [ask](#) — we will tell you what we know, or that we do not know yet. The [MDM Resource Centre](#) carries the guides and downloads.

What MDM does not do

Worth saying plainly, since assuming otherwise costs more time than reading a sentence.

MDM does not send MIDI Notes. Program Change, Bank Select and Control Change move patches and settings; Notes are a different message MDM was not built to send.

MDM does not edit sounds. It recalls what your equipment already has and organises how you reach it — it has no synthesis or effects parameters of its own to change. Your equipment's own editor remains the tool for shaping a sound; MDM starts once the sound already exists.

MDM does not record. Nothing you play through it is captured or played back.

MDM does not support MPE. Standard MIDI channels and messages only.

None of this is a gap being fixed quietly — it is the actual shape of the application, stated so you are not left guessing.

6. A word about the word “index”

MDM does not collect your patches. It collects their addresses.

A patch on your equipment is made of hundreds of settings — the amp, the effects, the levels, everything that makes it sound the way it does. Those settings live in the equipment and they stay there. A Searchable Instrument Index does not store them and could not recreate them: what it holds is where a patch is, not what it is made of.

What MDM stores is where a patch lives: its location, its name, and the MIDI messages that recall it. That is an index in the ordinary sense — the back of a book, not the book.

It is why the term is **Searchable Instrument Index** rather than library. A librarian keeps the books. MDM keeps the shelf numbers — and adds your own categories and notes, so you can find a patch by searching for “Sunday worship” or “verse clean” rather than remembering it was patch 47.

Most devices already come with an editor or librarian from their manufacturer, and MDM does not replace it. Those tools are for creating and editing patches, and they remain the best tools for that. MDM begins after the patch exists. It supplements them with something they do not offer: a fast way to find the patch you want and set it on your equipment from a Pad while you are playing.

One older word survives on screen. The **Patch Libraries** screen keeps its name, because that screen genuinely does behave like a library — it is where your indexes are kept.

7. Status colours, and what they are telling you

MDM shows a coloured indicator against each Device. It reports one thing only: **what MDM knows about the identity of the equipment at the end of that connection.** It is not a report on whether MIDI is working.

Green. MDM asked, something answered, and the answer matched what it expected. The Device is confirmed.

Orange. Something is not as it should be, but MDM can still send. Either the identity did not reconcile — a different manufacturer than expected, or more than one thing replying on the same connection — or the Device’s own port has gone and MDM has fallen back to the global port instead.

Red. The assigned port has gone and there is no global port to fall back to. Nothing is reaching that equipment at all. Red is the narrower case and the more serious one.

MIDI Identity not confirmed. MDM asked and nothing came back. For a great many devices this is simply the truth and will remain so for as long as you own them.

The fallback is worth understanding, because orange does not mean silence. MDM keeps one output port and one input port as a general default, chosen in Settings. If a Device’s own port disappears — an adaptor switched off, a cable pulled — MDM carries on sending, on that default. It tells you so, on the Device and on the badge.

Which matters because the Device is still sending, and what receives those messages is whatever is on the default port. On a rig where several pieces of equipment share an interface, something else may be receiving patch changes meant for the one that dropped — audible, and not at all what you expected. Silence is obvious and harmless. This is neither.

And here is the part that catches people out. A green Device can be completely silent, and an unconfirmed Device can work perfectly.

Green means MDM can reach the equipment and heard a sensible answer. It does not mean the MIDI channel is right, that the equipment is listening on that channel, or that a manufacturer setting somewhere is not quietly discarding everything that arrives. If a Device is green and your Pads do nothing, the MIDI channel is the first thing to check, not the connection.

And a device reading “MIDI Identity not confirmed” is not broken and not misconfigured. It is a device that does not answer that particular question. Send it a Program Change and it will change patch.

Since Build 107, MDM does not wait for silent devices when it starts. It asks, records the answer or the absence of one, and opens. It keeps asking quietly afterwards — so a device that is merely slow to reply, which happens on Bluetooth, will turn green by itself a little later.

From Build 108, this extends further. A device that goes properly silent — not merely slow, but unreachable for a while — no longer stays orange until you intervene by hand. MDM keeps trying on a slower cadence, and a device that comes back usually confirms itself without you doing anything. A related fix covers a different case: if iOS assigns the same named port a new internal ID — after a reconnect, or a system update — a Device used to show disconnected even though the port itself was still there. MDM now recognises it by name and reconnects on its own — but only when exactly one port matches that name. If more than one does, on an interface with several similarly-named ports, MDM declines rather than guess, and the Device stays on its visible fallback state until you resolve it by hand.

8. When an interface answers for your equipment

This is different from what happens during a scan, covered in section 4 — there, an interface's reply is simply listed like any other candidate, and recognising it is yours to do. This section is about a Device you already have: one already added and confirmed, where an interface has since begun answering in its place.

If MDM finds that an interface is replying to identity requests on behalf of the equipment behind it, it does not guess and it does not silently accept. It marks the Device and asks you.

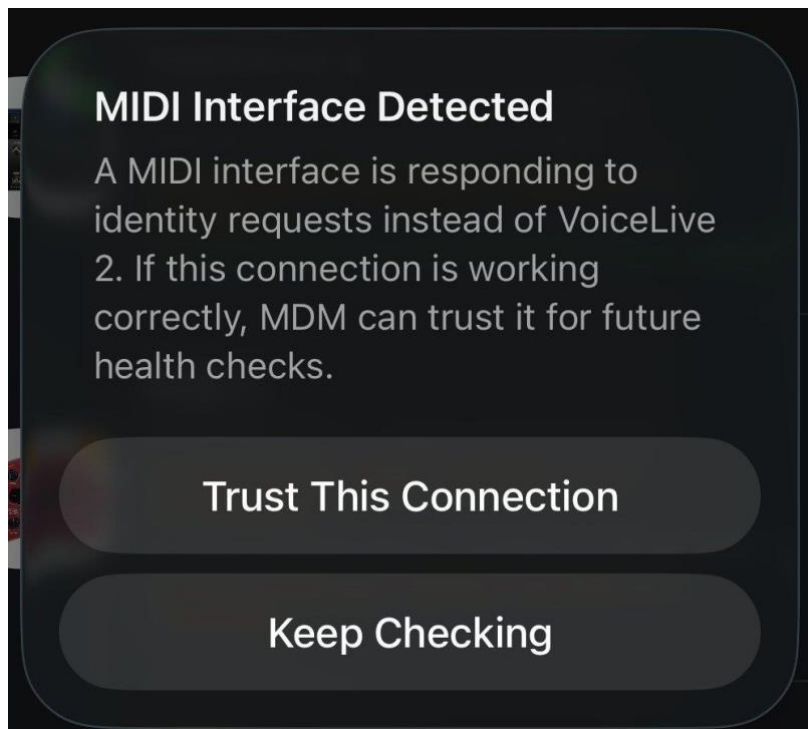


Figure 1.4 — MDM has noticed an interface answering for a device, and asks rather than assuming.

Trust This Connection tells MDM that this is your setup and the interface's reply is expected. MDM accepts that interface's replies from then on and the Device returns to green. The decision is remembered — it survives quitting the app, disconnecting the interface, and powering it off and on.

Keep Checking leaves things as they are. The Device stays marked and MDM keeps testing it. This is the right answer when you are not sure the connection is behaving, or when you would rather investigate than approve.

Trust can be withdrawn. From Build 107, if you trust an interface and later change your mind — you rewire, you replace the interface, you want to see the underlying state again — Edit Device lets you undo it.

Most musicians will never see this. It appears only when an interface answers identity requests itself, which in practice means a multi-port MIDI interface or certain Bluetooth adaptors.

9. Why MDM cannot tell you everything

There is a limit worth stating plainly, because knowing where it falls will save you time.

MDM can only report what reaches it. If a message leaves MDM, travels down a cable, arrives at a device, and the equipment discards it because of a setting in its own menus, MDM has no way to know. The port exists. The Device may well be green. The message went nowhere.

This is not a shortcoming in the application; it is a property of MIDI, which has no mechanism for a device to say “I received that and ignored it”.

Several manufacturers ship devices with a setting, enabled by default, that does exactly this — routing MIDI somewhere other than the sound engine, or requiring a USB control mode to be switched on, or

expecting a channel other than the one printed in the manual. Four different manufacturers were found to have one within a single day of testing, without looking for them systematically.

So: if a device is connected, shows a port, and does absolutely nothing, look in the equipment's own menus before looking at MDM. We keep a record of these settings for equipment we have tested. If yours is doing nothing and you cannot find the cause, [tell us what you have](#) — it may already be known, and if it is not, it should be.

10. What MDM does before it hands you the application

Everything in this part happens once, quickly, each time MDM starts. The opening screen is MDM finding out what is there before you touch anything.

It waits for your equipment rather than for a clock. Each piece is shown as it confirms, and the screen closes when they have settled — sooner on a simple rig, longer on a large one. It never closes before about eight seconds, because the application is also preparing itself behind it, and that takes a moment however little you have connected.

Equipment that does not answer no longer holds anything up. MDM asks, records that nothing came back, and opens. It keeps asking quietly afterwards, so something slow to reply — which happens on Bluetooth — turns green by itself a little later.

From Build 108, a device that goes properly silent, not merely slow, no longer stays orange until you intervene by hand. MDM keeps trying on a slower cadence, and it usually confirms itself again on its own. The same applies if the port itself changes identity underneath — a reconnect or a system update assigning a new internal ID to the same named port — MDM recognises it by name rather than treating it as missing, provided exactly one port matches that name. Where two or more do, on a multi-port interface, MDM declines to guess and leaves the Device on its fallback.

And it can tell you which connection is missing. MDM records the name of every MIDI port it has seen, so when an adaptor is disconnected it can say which one rather than only that something is absent.

A connection can only be named if MDM has seen that port before.

On the first launch after updating, a missing adaptor will not be named.

From the second launch onward it will. That is worth knowing rather than puzzling over: an unnamed absence on a first launch is the record being built, not a failure to recognise your equipment.

Nothing on that screen needs anything from you. It closes on its own, there is nothing to tap, and anything you need to act on will be waiting inside the application.

One habit worth forming if you use Bluetooth. Bluetooth adaptors do not always announce themselves when MDM opens. Tapping **Reconnect BLE MIDI** once at startup costs a second and removes a whole category of puzzling silence later. It is worth doing before you need it rather than after.

11. What to take from this part

- MDM sees ports, not devices. What is at the far end is discovered, or assumed, or told to it by you.
- Sending and listening are separate. MDM can send perfectly to something it cannot hear.
- Not every device answers an Identity Request, and some interfaces answer instead of it. Both are normal.
- A scan finds what answers. **What it does not find is not thereby incompatible.**
- Not everything that answers should be added. Interfaces and adaptors are the road, not the destination.
- Adding a Device by hand is one of two normal routes, not a repair.
- Green means MDM heard a sensible answer, not that MIDI is working. Check the channel next.
- “MIDI Identity not confirmed” is a permanent and harmless state for a great many devices.
- MDM cannot report a message a device silently discarded. When nothing at all happens, check the equipment's own settings.
- A missing connection can only be named if MDM has seen that port before — not on the first launch after updating, and from the second onward it can.

Part Two

Getting Your Equipment In

Part One explained what MDM can see. This part is what you do about it.

Before you start

Part One established that MDM sees ports rather than equipment, that not everything answers when asked, and that green does not mean MIDI is working. This part turns those facts into decisions: which route your equipment needs, what to do when a scan comes back empty, and what to check — and in what order — when a Device looks perfect and does nothing.

This part contains no procedures. The [Quick Start Guide](#) carries the steps for scanning, adding a Device and setting channels; this part explains which of those you need and why.

1. Choosing how your equipment gets in

Before anything else, one question decides most of what follows: how is this device going to reach your iPad or iPhone?

Does it have a USB socket that works with an iPad? If it does, a cable is the simplest route and the most reliable. Not every USB socket qualifies — a device that is not class-compliant has a USB port that will never appear, no matter what you plug it into. There is no setting for this and no way to tell by looking.

Two pieces of equipment cabled to each other will not talk, even with matching ports. Every USB connection needs one side acting as Host — in charge of the connection — and the other as Client, waiting to be addressed. Your iPad or iPhone is normally a Client itself, the way it behaves connected to a Mac. The Camera Adapter changes that: it puts your iPad or iPhone into Host role, and only then can a class-compliant piece of equipment be recognised at all. Two pieces of equipment cabled directly to each other are both Clients, each waiting for a Host that neither of them is, and nothing happens — not a fault, just nobody in charge of the connection.

If you connect it and no port appears, check the cable first. A USB-C cable sold for charging may have no data conductors and will carry no MIDI whatever. Once you have ruled the cable out and no port has appeared, the socket is not one an iPad can use, and the 5-pin sockets are your route.

Does it have 5-pin MIDI sockets? Then a MIDI interface or a Bluetooth adaptor will reach it. This is the most dependable route we know of, and for a great deal of equipment it is the only one — 5-pin MIDI has been the common language since 1983 and almost everything that speaks MIDI at all speaks it here.

Both? Either serves. A wired connection is not subject to the interference and range limits that make a wireless one drop, so it is the safer choice where you are not moving. Bluetooth is for when you want to stand further from your iPad than a cable allows, and it carries everything MDM does — including building an index — without difficulty.

Neither? A device with no MIDI sockets and no working USB cannot be reached by anything, and that is a property of the equipment.

One caution about buying an adaptor. Not every device can be hosted by every Bluetooth adaptor — some equipment advertises a USB speed that small adaptors cannot host, and no firmware update will change it.

Which adaptor, if you are choosing one. MDM is developed and tested every day with CME WIDI adaptors, and they are what we would point you towards. The reason is speed rather than loyalty: they are the fastest adaptors we have worked with, on our own equipment, and speed matters most when you

are building a Searchable Instrument Index — several hundred replies arriving one after another, where a quicker link is less likely to lose any of them. For ordinary playing, where recalling a patch is a single short message, the difference is invisible.

Other adaptors work perfectly well and a great many musicians use them without trouble. We simply cannot speak for all of them from our own use, and we would rather say so than imply a guarantee we cannot give. If you are weighing up a particular one, [ask](#) — we will tell you what we know, including when the answer is that we do not know yet. That is worth more before you buy than after.

2. When the scan finds nothing

You run Scan MIDI Studio, it completes, and your equipment is not in the list. This is the moment most likely to make someone conclude MDM does not work with their equipment. It almost never means that.

Work through it in this order.

- **Is there a port at all?** If MDM shows no port for the equipment, nothing has reached the scan. Check the cable first — a USB-C cable sold for charging may have no data conductors and will carry no MIDI whatever. A cable that states a data speed will work; one that advertises only watts may not.
- **Is the Bluetooth adaptor connected?** Bluetooth MIDI devices must be connected before MDM can see their ports. Connect them first, then scan. If you use CME WIDI adaptors, the [Quick Start Guide for those devices](#) covers pairing and the settings that matter — and much of it applies to Bluetooth MIDI adaptors generally, whoever made them. For other manufacturers, follow their instructions and then come back to MDM.
- **Is the equipment on the far side of an interface that answered for it?** Then it did appear — as the interface. See Part One, section 4.
- **Does the equipment answer Identity Requests at all?** A great many do not. If a port exists and the scan found nothing on it, this is the likeliest explanation by a distance.

In the last case, and it is the common one, there is nothing to fix. Add the Device by hand. It takes about a minute and the result is a Device that works exactly like a discovered one.

3. Adding a Device by hand

The [Quick Start Guide](#) carries the steps. What follows is what the fields mean and which of them matter.

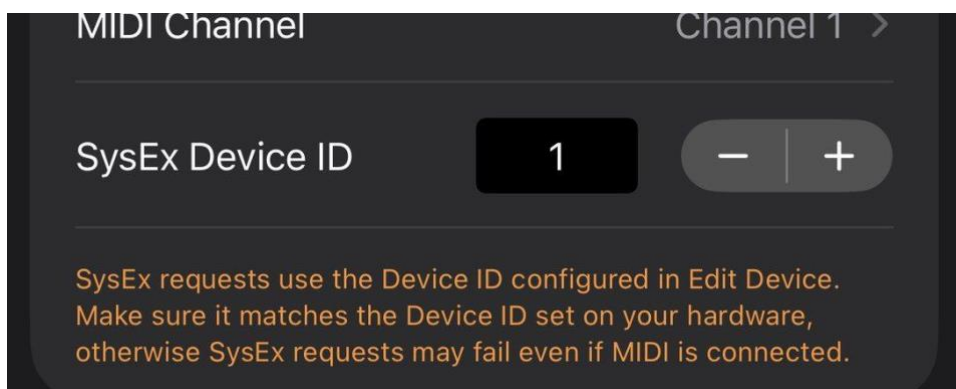


Figure 2.1 — The two fields that decide whether anything works: the MIDI channel, and the Device ID for SysEx.

Device Name is yours. Call it what you call it on stage.

Manufacturer and Model are optional. They do not affect whether MIDI works. The screen puts it plainly: they help MDM suggest compatible patch libraries later. Worth filling in — but an empty Manufacturer field will not stop a single message reaching your equipment.

MDM may offer to treat two manufacturer names as one. A manufacturer field filled in automatically from a real identity reply can read differently from one you typed by hand — "Roland Corporation" against "Roland", for instance. Where this happens, MDM asks: Same as an existing manufacturer, or Keep Separate. It never decides on its own, and either way nothing about the Device itself changes — only how its manufacturer is displayed and grouped. You will see the same offer in Add Device, in Edit Device, and as a banner in Settings for anything already sitting unresolved.

In Use switches a Device off without losing anything about it. Turn it off — in Edit Device, in Add Device, or with the quick toggle right on the Devices list — and nothing sent to that Device goes out at all; MDM reports why rather than staying quiet about it. Its Pads, Scenes and assigned ports stay exactly as configured, untouched, and the Device stops being checked for connection problems while it is off, so equipment that is genuinely not on the rig tonight never shows as a fault. Switch it back on and everything works immediately, with nothing to reconnect or set up again.

MIDI Channel is the field that matters. It must match the channel the equipment is listening on. This is the single most common reason a new Device does nothing, and it is covered in section 5.

SysEx Device ID matters for a narrower set of things. Building a Searchable Instrument Index, identity checks and other SysEx features use it, and it is a different number from the MIDI channel. If it does not match the hardware, Program Change will still work perfectly while every SysEx feature quietly fails. MDM warns about this on the screen itself.

Sliders and Buttons are set up here too. Below the fields covered above, Edit Device carries a **Sliders & Buttons** section of its own — separate from the fields that affect whether MIDI works at all. Part Four, section 3 covers what they do and how to use them.

The ports are the connection. Choose where MDM sends, and where it listens. On a simple rig these are the same. Where they differ, MDM will show both.

4. Reading a Device at a glance

Once Devices exist, the list tells you a great deal in a small space — provided you read the line, not just the dot.

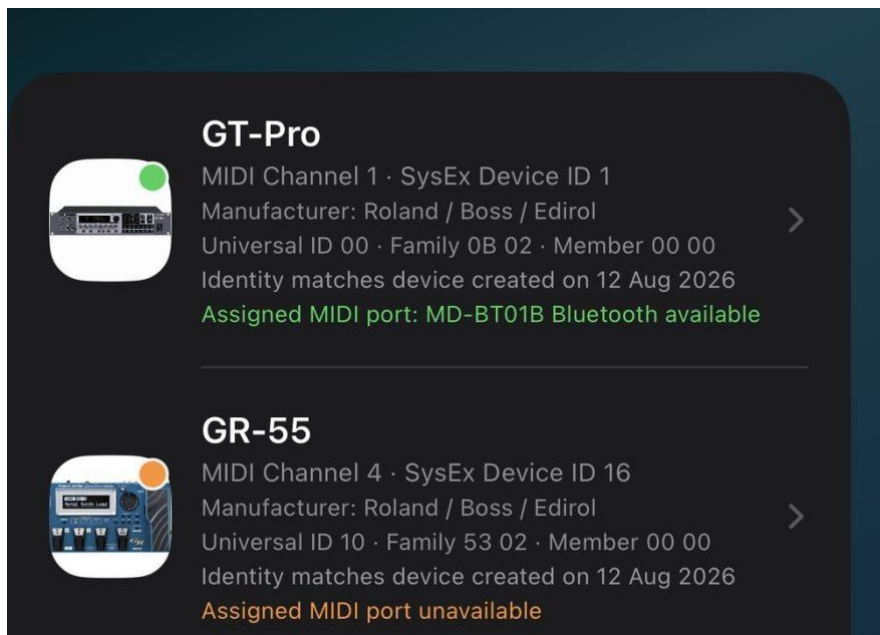


Figure 2.2 — Two Devices. One colour, and words the dot alone cannot carry.

The dot and the line share one signal, not two. They are never computed separately, and they never disagree — a Device with a green dot always carries green status text beneath it, and the same is true for orange. There is no colour that belongs to identity alone, and none that belongs to the connection alone.

What the dot cannot do is explain itself. It is a single colour with no words. The line can, because it carries an actual message — and that message, not a different colour, is what tells you which of several real causes you are looking at.

Orange is not one condition. It can mean an interface answered on behalf of the equipment — a question of identity, with a decision for you to make. Or it can mean the adaptor has been switched off and MDM has fallen back to the global port — nothing to decide, and nothing will reach the right equipment until it is back. The colour is identical either way. The words are not.

In the figure, the second Device reads “Assigned MIDI port unavailable”. That is not a mystery to be investigated. Its adaptor is not connected — the line said so directly, in words the dot alone could never carry.

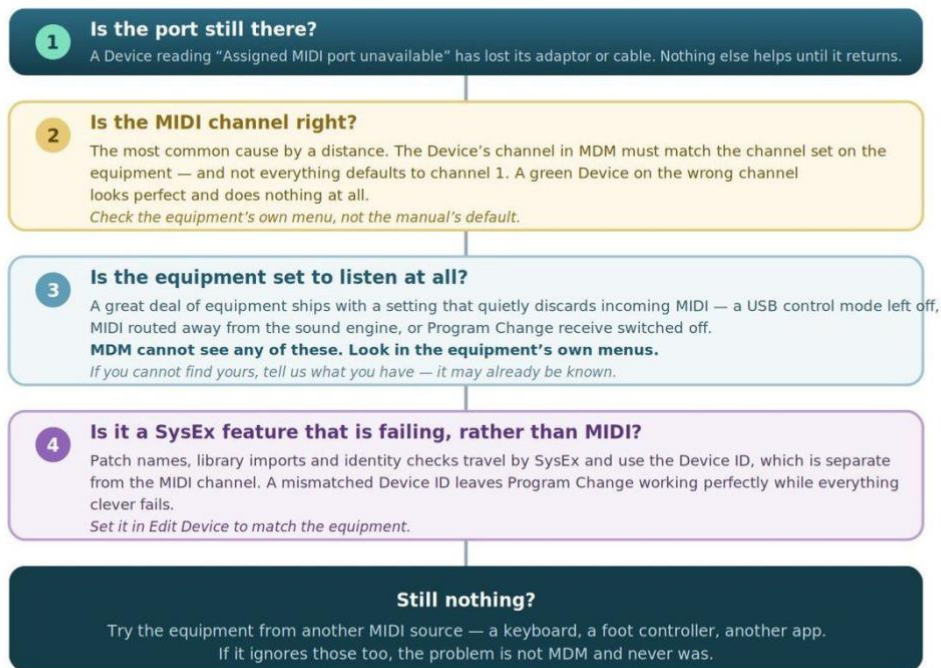
And the useful habit: read the line, not just the dot. A missing port explains everything downstream of it, and no amount of checking channels will help while it is missing.

5. When a Device is green and nothing happens

This is the case that sends people looking in the wrong place, because everything on screen says the connection is fine — and it is. Green means MDM reached the equipment and heard a sensible answer. It has never meant the equipment is listening to what MDM sends.

Nothing happens when you tap a Pad

Work through these in this order. Most cases stop at the second.



Green means MDM heard a sensible answer. It has never meant the equipment is listening.

Figure 2.3 — Four checks. Work through them in this order.

The order matters more than the list. Each check rules out a whole class of cause, and the first two account for most of what we have seen. It is the recommended order rather than a proven one — but a missing port makes everything below it meaningless, so it belongs first whatever else is true.

A note on channels, because it is worth stating plainly. Not every device defaults to channel 1, and a manual stating a default is not evidence of what your equipment is set to today. Read the channel from the equipment's own menu. A device on channel 10 with a Device on channel 1 is silent, and everything about that silence looks like a fault in the application.

A note on devices that discard MIDI on purpose. Several manufacturers ship a setting, on by default, that stops incoming MIDI reaching the sound engine — a USB control mode that must be switched on, MIDI routed to a different destination, or Program Change receive switched off. MDM cannot detect any of them, because nothing comes back to say so. When a device ignores everything and the channel is right, this is where to look. We record these settings for equipment we or our testers have worked with; if yours is not among them we will say so rather than guess. [Tell us what you have](#) either way — an answer we do not have yet is the one most worth adding.

And if you have worked through all four and the equipment is still silent, try it from another MIDI source entirely — a keyboard, a foot controller, another app. If it ignores those too, the problem is not MDM and never was.

6. What to take from this part

- A USB socket that does not appear is not a fault to fix. Use the 5-pin sockets and an interface.

- A cable is the safer choice where you are not moving; Bluetooth carries everything MDM does, including building an index.
- A scan finding nothing usually means the equipment does not answer. **Add it by hand and carry on.**
- On the Add Device screen, only the MIDI channel and the ports affect whether MIDI works. Manufacturer and Model are for finding libraries later.
- The Device ID is not the MIDI channel. A wrong Device ID breaks SysEx features while Program Change works perfectly.
- Orange is not one condition — an identity that did not reconcile, or a port that has gone and fallen back to the global default. Read the status line to find out which.
- Red is the serious one: the port has gone and there is no fallback, so nothing is reaching that equipment.
- A missing port explains everything below it. Fix that before checking anything else.
- Green and silent almost always means the MIDI channel — and after that, a setting inside the equipment.

Part Three

Patches by Name

Finding the patch you want, instead of remembering where you left it.

Why bother

You can use MIDIDeviceManager perfectly well without ever building an index. Enter a Program Change number on a Pad, name the Pad whatever you like, tap it, and your equipment changes patch. Everything works, and the Pad says Verse Clean rather than 46.

What that costs you is typing, and searching. Every name is one you entered by hand, and it exists only on that Pad. Nothing else in MDM knows the patch is called anything — so you cannot search for it, filter for it, or find it again anywhere but the Pad you put it on.

A Searchable Instrument Index saves you the typing and adds the searching. The names arrive already attached to the addresses, so you choose a patch by name instead of entering a number and labelling it yourself — and once they are in an index, every one of them can be searched and filtered. Part One explained what an index is: the addresses of your patches, not the patches themselves. This part is about getting one, and about the part most people never reach — making it find what you actually meant.

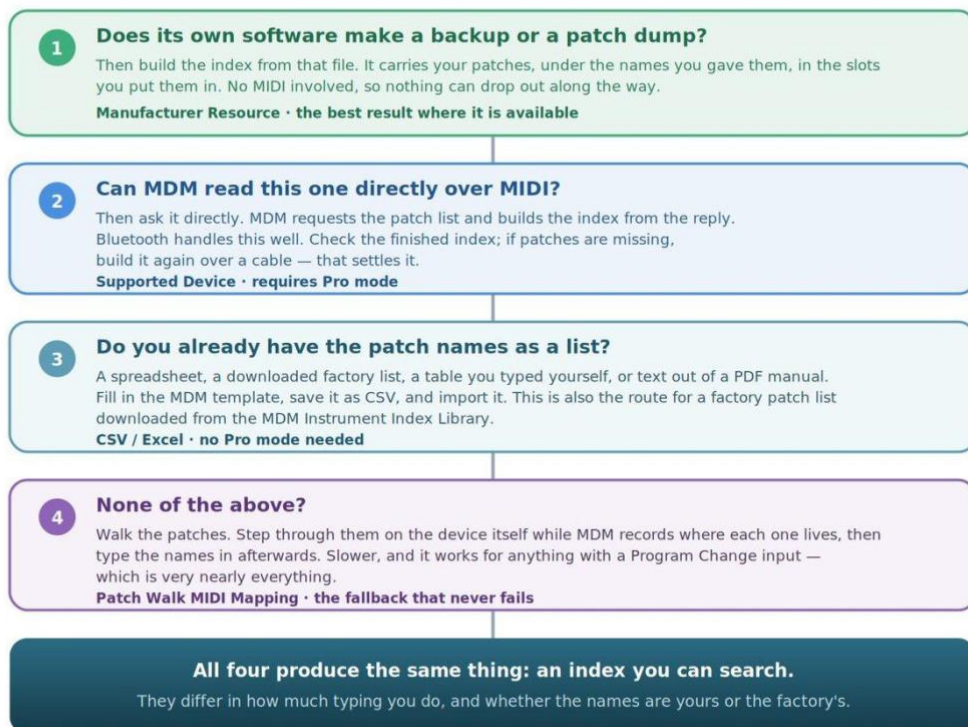
This part contains no procedures. It explains which route applies to your equipment and what an index is worth once you have one. The [main Quick Start Guide](#) and the [guide to importing an index](#) carry the steps.

1. Four ways in, and which one is yours

There are four ways to build an index, and they are not alternatives to weigh up. They are a list to work down: take the first one your equipment allows.

Four ways to build a Searchable Instrument Index

Work down the list. Take the first one your equipment allows.



Every route is on one screen: Libraries → Import Device Index. The Quick Start Guides carry the steps.

Figure 3.1 — Work down the list. Take the first one that applies.

Why in that order. The higher routes need less from you and give a better result. A file from your equipment's own software already knows your patch names and where they live. Walking the patches by hand gets you to the same place, but you type the names in yourself.

The difference that matters most is whose names you end up with. An index built from your own device contains your patches, under the names you gave them, in the slots you put them in. Rename a patch or reorganise a bank, build the index again, and it stays correct. A published factory list can never do either — it is right about the equipment as it left the factory, and wrong about yours the moment you change something.

That is not an argument against factory lists. For equipment you play on factory patches, a downloaded list gets you searching in two minutes and costs nothing. Ready-made indexes for a growing number of devices are published on the [MDM device compatibility page](#). It is only worth knowing which kind you have, so that a name not matching the patch is not a mystery.

2. All four routes live on one screen

Everything in Figure 3.1 is reached the same way: **Libraries**, then **Import Device Index**. The screen names the four routes and says what each one needs. The [Manufacturer Resource Quick Start Guide](#) walks through the first route in full. The [Importing a Searchable Instrument Index Quick Start Guide](#) walks through the CSV route and the direct-from-equipment route in full.

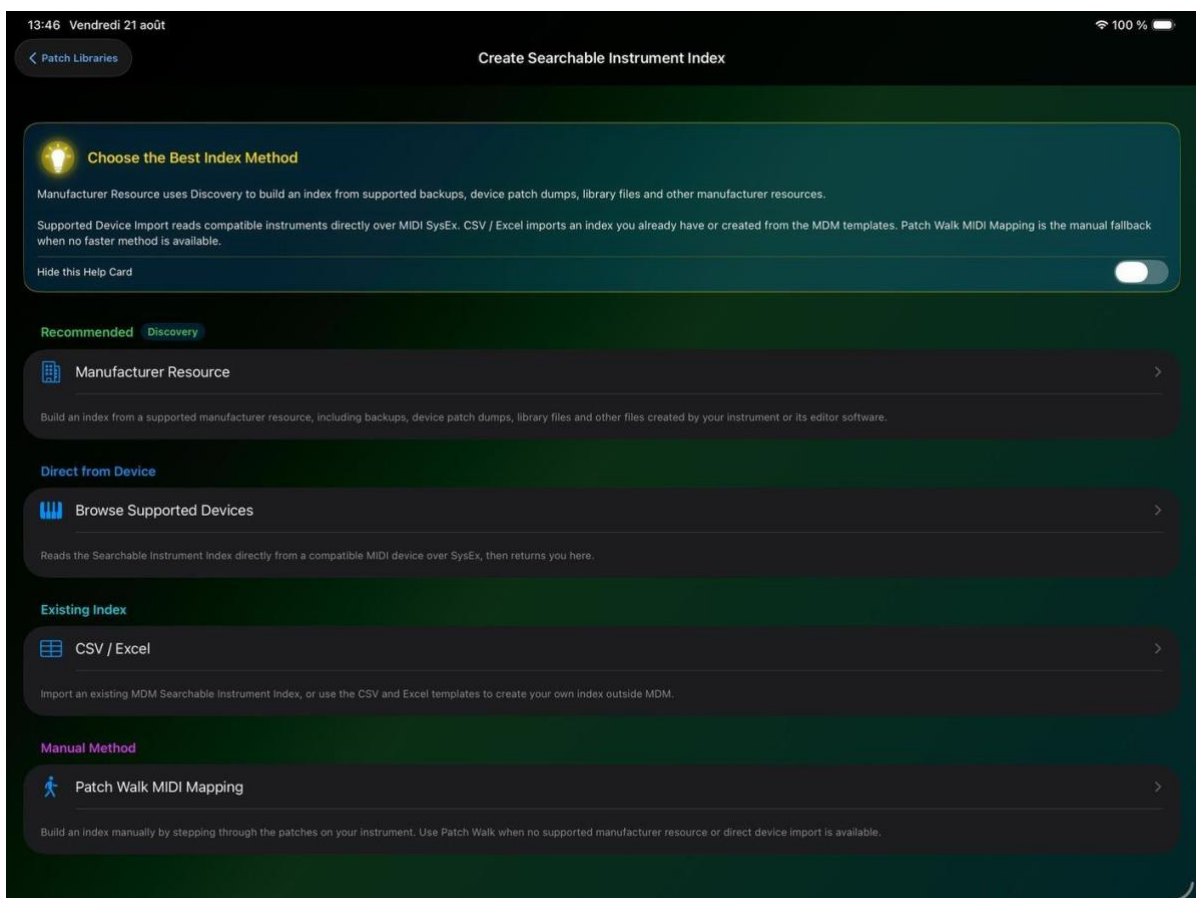


Figure 3.2 — The four routes, as the application presents them. Shown here on iPad.

Browse Supported Devices is available in both Standard and Pro. Both see the identical simplified screen through this route — Target and Tests, nothing more. If you are in Pro mode and want the full SysEx Lab, it is reached separately, from Tools, not from here.

Which equipment can be read directly is a list that grows. Each build usually adds one or two, and the current list is published on the [MDM device compatibility page](#) rather than here — so that it is right on the day you read it rather than on the day this was written. If yours is not on it yet, one of the other three routes will still get you an index.

One thing to know about the second route. Reading a patch list over MIDI means several hundred replies arriving one after another. Bluetooth carries this perfectly well — every device MDM can read directly has been imported over an adaptor, complete and without skips, including three sharing one. But any wireless link can in principle drop a reply, and the symptom is worth recognising: the import completes and looks successful, and patches are missing from the finished index.

So check the finished index rather than assuming, and if patches are missing build it again over a cable. That settles it either way. There is no need to reach for a cable pre-emptively.

3. What an index is worth once you have one

The obvious gain is names instead of numbers. The larger one arrives later, and it is the reason the word is searchable rather than simply index.

You can find a patch by name. Type part of what you remember and MDM narrows the list as you go. This alone removes most of the reason people keep a written patch list beside the equipment.

Sometimes the name you get is better than the one your equipment shows you. Some equipment displays only a slot number — 1A, 2B — and never says what is in it. Where MDM can read something useful out of the patch itself, it adds it: on the Line 6 POD 2.0 the amp model is read and appended, so an entry reads “1A · Big Lead Tone (Treadplate)” rather than a bare number. If you find a detail in brackets after a patch name, that is where it came from.

You can assign a patch to a Pad without knowing its number. Choose the patch; MDM fills in the Program Change and Bank Select values for you. If several names look similar, arrows beside the patch name step through the list and send each one to your equipment as you go, so you can hear what you are choosing before you commit it.

And you can search on things the manufacturer never thought of. This is the part worth reading twice.

4. Making the index find what you meant

A factory patch name tells you what the designer thought the patch was. It rarely tells you what you use it for.

An index can carry your own categories and notes alongside each patch, and those are searchable in exactly the same way the names are. So you can write against a patch whatever would actually help you find it again:

- the song you use it for — “Hotel California”, “Sunday worship”

- what it does in the arrangement — “lead solo”, “verse clean”, “big pad”
- your own honest opinion — “favourite clean sound”, “too bright live”
- who it is for — “Dave's cue”, “second set only”

Then search for those words instead. MDM finds the patches you tagged, whatever the factory called them. A patch named U04-2 Rotary Poly Key is impossible to remember and trivial to find once you have written “verse organ” against it.

This is what turns an index from a convenience into something you rely on. Names alone help you when you half-remember a patch. Your own notes help you when you remember only what you needed it for — which, three months and forty songs later, is usually all you remember.

Build the notes as you go. Adding a note the moment you find a patch you like costs a few seconds. Trying to reconstruct them across a whole device in one sitting is an afternoon nobody enjoys.

5. Knowing which patch you are on

Recalling a patch and knowing which patch you are on are different problems, and an index solves only the first. The second has its own answer, and it is a better one than most people expect.

MDM has two ways to find out what your equipment is currently set to.

It can ask. A SysEx request, where your equipment supports one and a request has been configured for it. This is the route that depends on SysEx, and a great deal of equipment does not implement it.

Or it can listen. When you change patch on the front panel, a great many devices announce it by sending a Program Change. MDM can watch for that and capture it straight into a Pad. No SysEx involved at any point — though not everything announces, and equipment that says nothing cannot be listened to.

If your equipment does not answer SysEx, listening is your route.

Running the probe once will tell you whether yours announces.

This matters more than it sounds. A good deal of equipment never answers a SysEx request of any kind, and a musician with such a unit could reasonably conclude that MDM will never know what they are set to. That is not the case. Announcing a patch change is ordinary MIDI behaviour and a great many devices do it — but whether yours transmits depends on a setting in its own menus, and plenty leave it off as they leave the factory. Running the Passive MIDI Monitor probe once settles it for your equipment, which is worth more than any general claim about equipment at large.

Two things have to be true. Your equipment must be set to transmit Program Change — a setting in its own menus, and switched off on plenty of units as they leave the factory. And MDM must know that it does, which comes from running the Passive MIDI Monitor probe once. After that, the Pad editor offers to listen.

Listening gives you the address, not the name. Bank and program number arrive; the words do not. The name comes from an index, if you have one. So the two features complement each other rather than competing — listening finds where you are, and an index tells you what it is called.

And where neither is available, MDM genuinely cannot tell you. Equipment that answers no SysEx request and transmits no Program Change sends nothing back at all. There is no third route and nothing to read.

6. What an index does not do

Two edges are worth stating plainly.

An index does not change your equipment. It reads where things are; it never writes. Building one, rebuilding one, or deleting one has no effect on a single patch in your equipment.

And an index does not replace your equipment's editor. As Part One put it: the manufacturer's editor makes the patches, and remains the best tool for it. MDM starts after the patch exists, and takes over the part the editor was never for — finding it in a hurry, on stage, under lights.

7. What to take from this part

- Four routes, in order. Take the first one your equipment allows.
- An index built from your equipment carries your names. A factory list carries the factory's.
- All four routes are on one screen: Libraries → Import Device Index. The CSV and Excel templates are downloaded from that screen, not from the website.
- All four routes are open to both Standard and Pro, and the direct-from-device route looks identical in both — the full SysEx Lab is a separate destination, reached from Tools, not part of this flow at all.
- Bluetooth handles an index build well. Check the finished index, and if patches are missing **build it again over a cable** — a partial import still looks like a successful one.
- Your own categories and notes are searchable. They are the difference between an index that is useful and one you depend on.
- Write the notes as you go, not all at once.
- An index never changes your equipment, and never replaces its editor.
- If your equipment does not answer SysEx, **run the Passive MIDI Monitor probe once** — MDM can then watch for the Program Change your equipment sends and capture it into a Pad.

Part Four

Building a Performance

Turning a room full of equipment into something you can play.

What this part is for

By now MDM can reach your equipment and, if you have built an index, set patches on it by name. This part is about the layer above that: turning those patches into something you can operate while you are playing, when your hands are busy and there is no time to think.

The [Quick Start Guide](#) shows how to make a Pad and a Scene. This part explains which of them you actually want, why a performance built badly is harder to fix on stage than one built simply, and — for Pad organisation and Setlists — how to do it.

Two procedures live here rather than in a guide: organising a Pad grid, and building a Setlist. Both are things a musician returns to as the rig changes rather than tasks completed once. Everywhere else, the guide that carries the steps is named with the task.

1. Three layers, and choosing between them

Almost everything in a performance environment is one of three things.

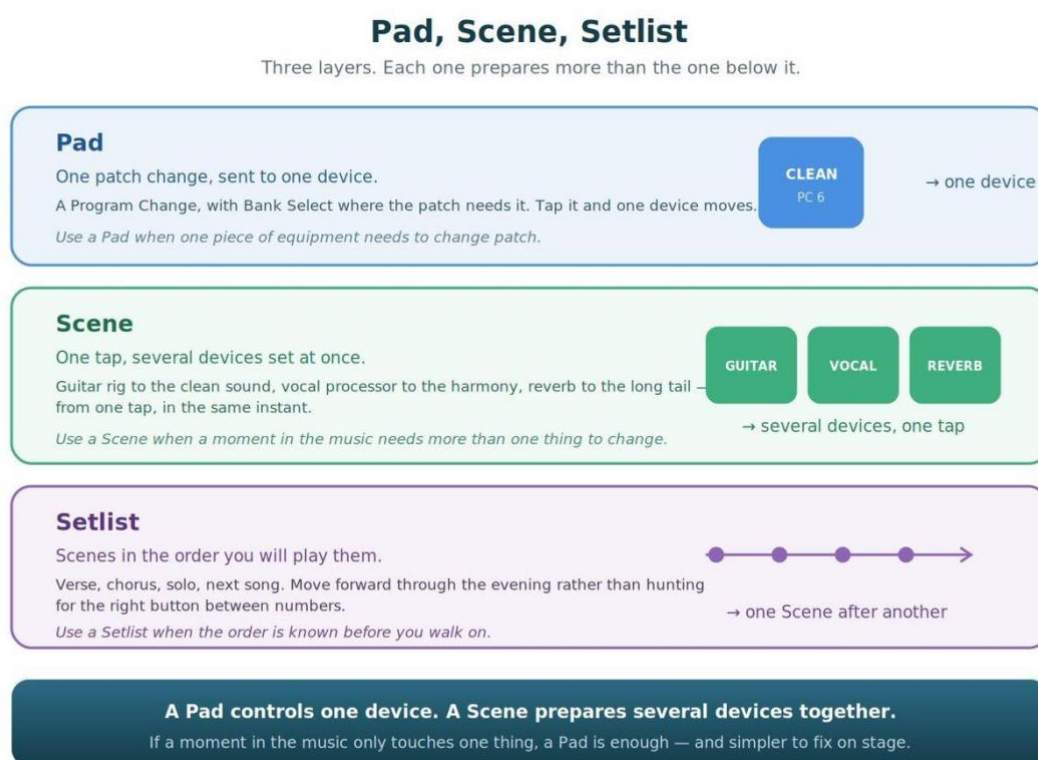


Figure 4.1 — Each layer prepares more than the one below it.

A Pad controls one device.

A Scene prepares several devices together.

That sentence decides most cases. If the moment in the music only touches one piece of equipment — the guitar rig changing patch between verse and chorus, and nothing else changing — a Pad is enough.

A Scene is for a moment, not for a patch. When the chorus arrives and the guitar rig, the vocal processor and the reverb all need to be somewhere different, that is a Scene. One tap and everything moves — quickly enough to be one action as far as the music is concerned — and nothing is left behind because you ran out of hands.

The mistake worth avoiding is building Scenes for everything. A Scene that prepares one device is a Pad with more moving parts: more to set up, more to check before a gig, and more to go wrong under pressure. Simplicity on stage is worth more than symmetry in the design.

And Setlists are for order. If you know before you walk on what follows what, a Setlist puts them in that order so you are not searching a long list. If your evening changes shape as it goes — requests, a shortened set, a song dropped — Scenes on their own may serve you better.

2. Pads worth having

A Pad is the smallest unit of a performance and the one you will make most of. A few habits make them worth having under pressure.

Name a Pad for what it does in the music, not for what the patch is called. A factory patch name tells you what the designer thought. On stage, **Verse Clean** is more use than **U04-2 Rotary Poly Key**, even though they are the same patch. You can always find the original name in the index.

Test the Pad when you make it, not before the gig. Tapping a Pad sends its message immediately. The moment to find out that a value is wrong is while you are sitting down with the equipment in front of you.

A Pad sends a patch change. A Program Change, preceded by Bank Select where the patch needs it — and, if you choose to configure one, a single fixed Control Change alongside it. The Control Change is optional and most Pads will never carry one; where a Pad does, it is set once in Edit Pad under Also Send, and sent exactly the same way every time you tap the Pad — not varied, not tracked, the same value alongside the same patch change.

The Triggered by settings in the Pad editor point the other way. They are not what the Pad sends — they are what makes the Pad fire. A footswitch sending a note, or a Control Change from a backing track, can trigger a Pad; the Pad then sends its patch change as usual. The labels say **Triggered by** for exactly this reason: they were renamed once it became clear how easily they read as outgoing.

Organising a Pad grid

Once a device has more than a dozen Pads, finding one becomes the problem rather than making one.

Colour earns its place on a mixed grid. A Device that drives transport controls — Start/Stop, Rewind to Beginning — alongside ordinary song Pads can leave the two indistinguishable at a glance, once both sit in MDM's automatic colour scheme. Edit Pad's own colour picker fixes this directly: colour the transport controls one way and the songs another, and the grid separates itself before you even read a label. One tap on Use Default Colour returns any Pad to where it started.

Reorder lets you drag Pads into the order you want. The order is yours and MDM keeps it — it is not alphabetical unless you make it so.

Sort A–Z arranges them alphabetically in one action, and **Undo Last Sort** puts them back exactly as they were. That pairing is deliberate: you can try alphabetical order without losing the arrangement you spent an evening building.

Grouping and filtering change what you see, not what you have. Your manual order is preserved underneath. Turning a filter on to work on one part of a set does not rearrange anything.

The order that matters is the order you use. Alphabetical is convenient for finding something at home. During a song, the Pad you need next should be next to the Pad you just used.

3. Sliders and Buttons

A Pad changes the patch. Sliders and Buttons work alongside it, controlling something continuously or immediately without touching the patch itself — a level, an effect, anything the equipment accepts as a Control Change.

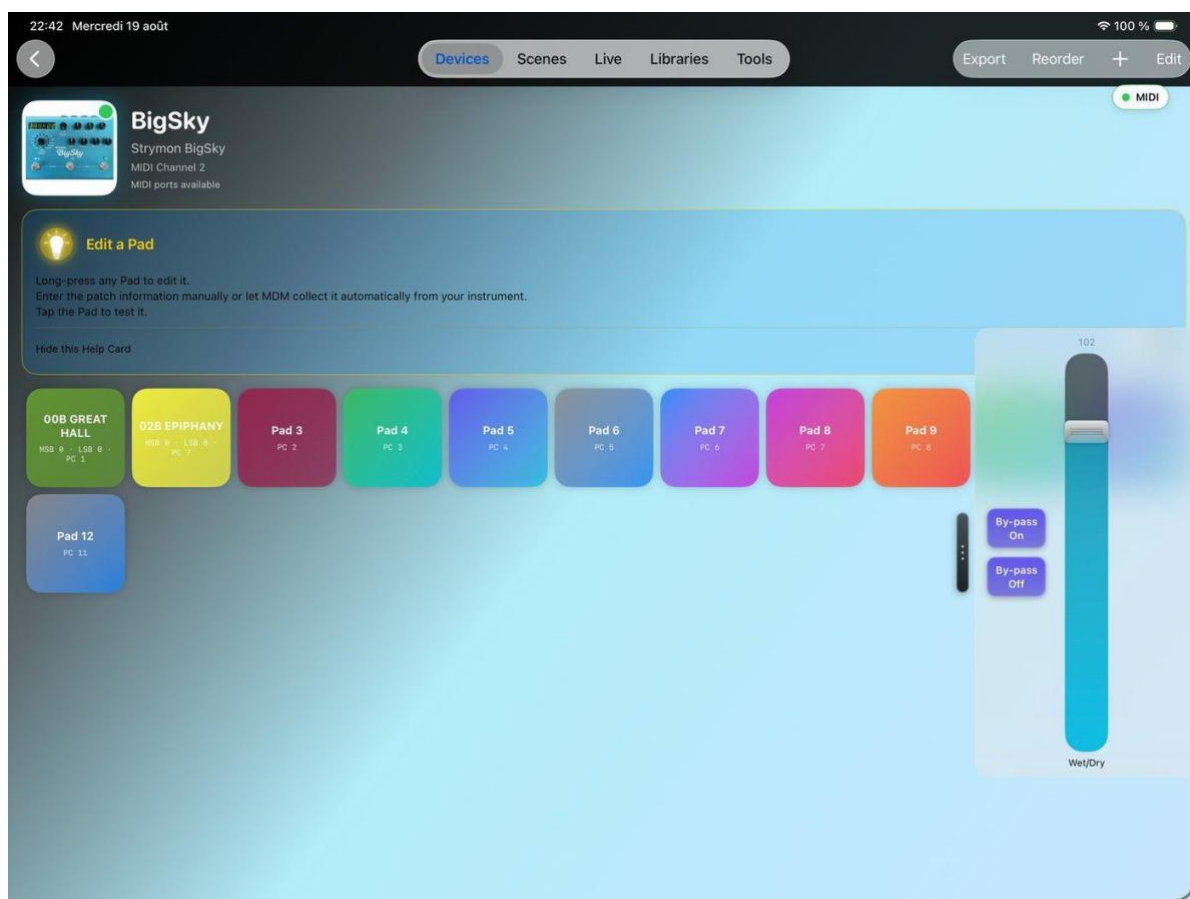


Figure 4.2 — A Slider and a Button pair, open beside a BigSky's Pads. The live value shows above the handle as you move it.

They are created in Edit Device, under **Sliders & Buttons** — separate from any Pad, and available regardless of which patch is currently selected. Once created, they stay in effect after a Pad recalls a new patch; changing the patch does not reset them.

A Slider sends Control Change across its full range, 0 to 127. You choose which CC number it controls, either from a General MIDI suggestion or typed in directly.

A Button is not one control with two states — it is a pair. Add On/Off Pair creates two independent buttons at once, sharing a name and a CC number but each with its own fixed value to send, defaulting to

127 and 0. Tapping either always sends its own value; nothing is tracked, and nothing alternates. MDM has no way to confirm what the equipment is actually set to, so rather than claim to toggle a state it cannot verify, it gives you two buttons with two known, fixed outcomes.

Where you reach them: a handle at the right edge of the screen, on Device Pads, on Scenes, and in Live. Tap it to snap the panel open to a standard width or closed again; drag it to any width in between. On Device Pads the panel always shows that Device's own controls. On Scenes and Live, it shows only the controls belonging to whichever Device the current Scene actually uses — not a permanent fixture there the way it is on Device Pads.

Name for the Device, not just the control, anywhere you use Scenes or Live. The panel shows only a control's own name — nothing beside it says which Device it belongs to. On Device Pads that is never a problem, since only one Device is ever in view. On a Scene spanning several Devices, two similarly named sliders on two different Devices are not distinguished by anything in the panel itself. *Chorus Level* is ambiguous the moment more than one piece of equipment has one; *GT-Pro Chorus Level* is not.

Channel follows the Device by default. Turning on Use Different MIDI Channel at creation lets a single Slider or Button send on a channel of its own — but unless you do, it uses whatever channel the Device is already set to. There is no limit to how many Sliders and Buttons a Device can carry, and no manual reordering: they appear in the order you created them.

4. Scenes

A Scene is one action that prepares several devices at once. It is the feature that turns a collection of equipment into a rig.

Build a Scene around a moment in the music. The name that works is the name of the moment — **Chorus 2, Solo, Song 4 Intro** — rather than a description of the settings inside it. You will be reading it at speed.

A Scene sends everything it holds, one message after another, faster than you could tap. From where you are standing it is a single action. It is not a sequence you have to wait through, and there is nothing to time.

Include the devices that need to change, and leave out the ones that do not. A Scene that sets nine devices when only three of them move is nine chances for something to be wrong, and it takes longer to check.

Channel 16 is reserved. MDM keeps MIDI channel 16 for Scene triggers, and never relays a trigger message to any Device. If your own MIDI setup routes a trigger source into the same path as a Device on channel 16 — a MIDI THRU box, a merge box, several instruments daisy-chained on one cable — that Device would receive it too, entirely outside MDM's control. Leaving channel 16 free means never needing to know whether your rig is wired that way. The [Quick Start Guide](#) covers this in its channel chapter.

5. Setlists

A Setlist is Scenes in the order you will play them. It exists so that the question during a performance is never “where is the next one”.

Build it in playing order, not in the order you made the Scenes. Including the gaps: if there is a talk between two songs, the Setlist should not make you skip past something to reach the next number.

A Setlist does not lock you in. A Setlist arranges Scenes; it does not step through them. Each one is recalled by tapping it or by its own trigger, which also means a song called for out of order is no harder to reach than one in sequence.

Build it before the day, and walk it once. Stepping through a Setlist at home with the equipment connected is the cheapest rehearsal there is, and it catches the Scene that was never finished.

6. Live Mode

Live Mode is the same performance, presented for playing rather than for building. Larger targets, fewer things on screen, and nothing that invites you to edit something by accident.

Set the rig up in Devices and Scenes. Play it in Live Mode. Keeping those two apart is what stops a mistimed tap changing your configuration mid-song.

Pad sizes are worth setting deliberately. The size that looks right sitting at a table is not always the size you can hit standing up, in the dark, in a hurry. MDM lets you set Live Pads and Device Pads separately, and it is worth spending five minutes on it before it matters.

7. Playing without touching the screen

Most performers would rather not look at an iPad while playing. MDM can be driven entirely from something else.

Anything that sends standard MIDI can trigger MDM. A foot controller, a keyboard, a drum module, a DAW, a backing track application, a show control system. MDM listens for a specific incoming message and fires the Pad or Scene attached to it.

Triggers are learned, not typed. In the Pad or Scene editor, under **Triggered by**, choose **Learn** and press the pedal. MDM captures what arrived and assigns it, so you never have to look up which message your controller sends. This works for anything, including equipment whose documentation you no longer have.

Many foot controllers send Control Change rather than Program Change, and MDM accepts either. If a pedal appears to do nothing, that difference is worth checking before anything else.

Give each trigger its own destination. A trigger fires a particular Pad or a particular Scene, so a controller with several buttons gives you several places to jump to — one per button. Assign the ones you reach for most and leave the rest of the evening to your hands.

A backing track application can drive the whole rig. If your tracks already send MIDI at the right moments, MDM can take those messages and move every device with them, and the performance runs itself while you play.

Back up what you have built

Tools carries Backup & Restore — one JSON file holding everything MDM knows: every Device, Scene, Searchable Instrument Index, Setlist, custom manufacturer and alias, and every Device's Sliders and

Buttons. Create Full Backup builds it; Export Device Backup does the same for one Device alone, with just its own assigned library, sliders and buttons.

Restoring the two is not the same action. A Full Restore replaces everything wholesale — whatever is currently on the iPad is overwritten by what is in the file, not merged with it. Importing a single Device adds it alongside what is already there instead.

This backs up MDM's own records — names, channels, ports, addresses — never the actual sound living inside your equipment, for the same reason a Searchable Instrument Index never does either. Restoring, on this iPad or a different one, matches each Device back to whatever is actually connected by name, manufacturer, channel and port, the same way MDM would confirm it on first use.

8. What to take from this part

- A Pad controls one device. A Scene prepares several devices together. That decides most cases.
- A Scene that prepares one device is a Pad with more to go wrong.
- A Pad can carry one optional Control Change alongside its patch change — set once, sent exactly the same way every tap.
- Sliders and Buttons live in Edit Device, separate from Pads, and survive a Pad changing the patch. Every Button is really two — an On and an Off — since MDM cannot verify what the equipment is actually set to.
- On Scenes and Live, name Sliders and Buttons for the Device as well as the control — the panel does not show which Device a control belongs to.
- Name Pads and Scenes for the moment in the music, not for the settings inside them.
- Sort A–Z is safe to try, because Undo Last Sort restores your own order exactly.
- Grouping and filtering change what you see, never what you have.
- Leave channel 16 alone — it belongs to Scene triggers.
- Build the rig in Devices and Scenes; play it in Live Mode.
- Learn a trigger by pressing the pedal rather than looking up what it sends.
- Each trigger fires one named Pad or Scene, so a multi-button controller gives you one destination per button.

Part Five

Playing

What changes when the room is full and it is your turn.

What this part is for

Everything so far has been preparation, done sitting down, with time to check. This part is about the hour when none of that is available: when you are playing, something else is on the screen, and whatever goes wrong has to be fixed in the gap between two songs.

Two things dominate. Keeping MIDI alive when MDM is not the app in front of you, and knowing what has happened when a piece of equipment stops answering.

The [Quick Start Guide](#) carries the steps for starting a session and setting the confirmation interval. This part explains what those things mean while you are playing.

1. MIDI while you are looking at something else

Almost nobody performs with MDM as the only thing on screen. Lyrics, chord charts and backing tracks all want the same display. There are two ways to have both, and on current systems you can choose between them.

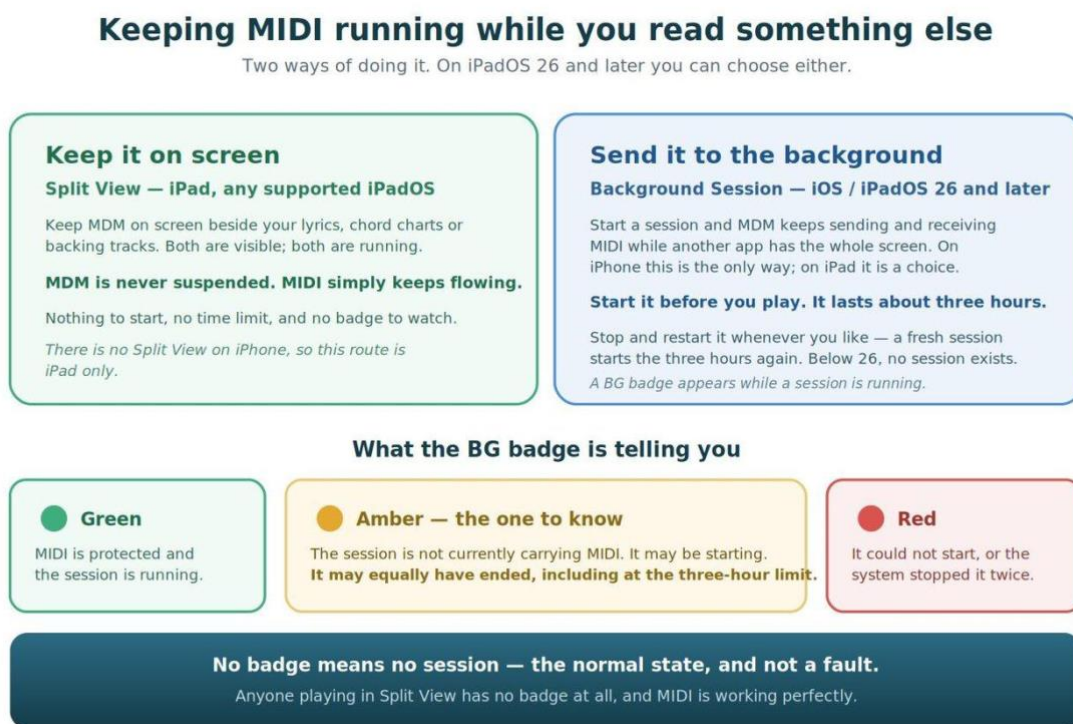


Figure 5.1 — Two ways to keep MIDI running, and one badge worth reading correctly.

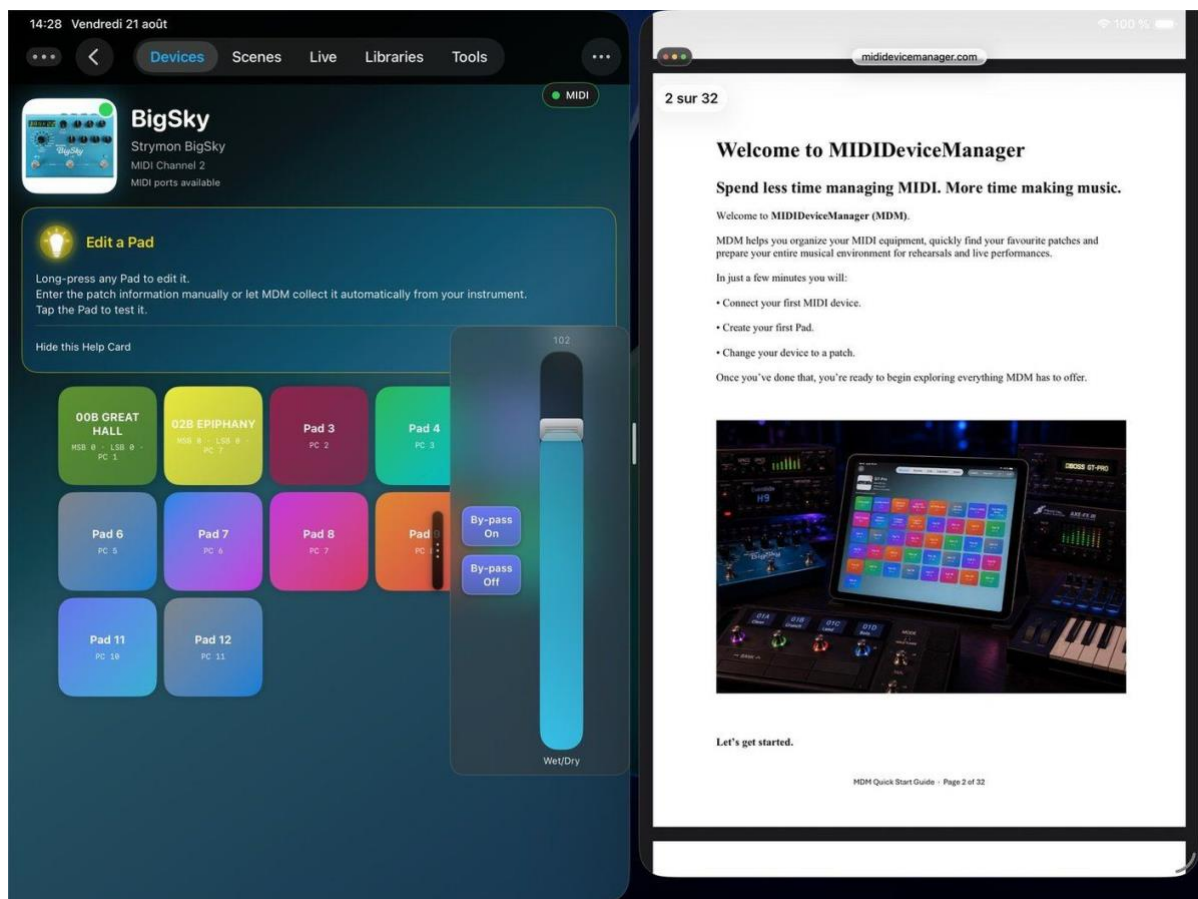


Figure 5.2 — Split View in practice: MDM on the left, the Quick Start Guide open on the right. Both fully usable at once.

Split View keeps MDM on screen beside the other app. Both are visible, both are running, and MDM is never suspended. There is nothing to start, no time limit, and no badge — which is why a musician playing this way sees no BG badge at all and everything works. This is iPad only; there is no Split View on iPhone.

A Background Session lets the other app have the whole screen. From iOS and iPadOS 26, MDM carries on sending and receiving MIDI with nothing of its own visible. On iPhone this is the only route. On iPad it is a choice, and a good many musicians prefer it — lyrics are easier to read at full width than in half a screen, and the BG badge is a reassurance that Split View does not give you. Below version 26 there is no session at all, and MDM must stay on screen to receive MIDI.

You can stop a session and start another whenever you like. A fresh session begins its three hours from scratch, so the sensible habit is to end one during a planned break and start a new one before the next set. That is far better than letting a session run down and discovering it has gone at the worst possible moment.

Start the session before you play, not when you need it. Starting it is a deliberate act, and the moment you discover you have not is the moment you cannot easily fix it.

2. A second surface: the Apple Watch companion

MDM now runs on the Apple Watch, and it stays in step with whatever is on your phone or iPad while you play. Tap a Pad on either device and it highlights on both — the Watch is not a second, separate copy of MDM, it is the same session, seen from your wrist.

The real value on stage is not the convenience — it is the second pair of hands. Because the Watch and the phone stay independently in sync rather than simply mirroring one another, they can be pointed at two different pieces of equipment at once. One hand on the phone recalling patches on a guitar processor, the other glancing at the wrist to recall something entirely different on a keyboard rig — two Devices, two surfaces, no juggling a single screen between them.

The Devices page on your Watch shows the Pads of whichever Device sits first in your phone's Devices list. This is not a limitation to work around — it is how you choose. Reorder your phone's Devices list and the Watch follows, so you always know exactly which equipment your wrist is controlling: whichever one is at the top. Scenes are not affected by this, since a Scene is already built to reach several Devices at once.

Two things are worth knowing before you rely on it. The very first tap of a session, after a period of inactivity, does not always register on the Watch immediately — every tap after that one works reliably, so a second tap or simply moving on to the next Pad is all it takes. And if the Watch shows "iPhone unreachable," this is almost always the Watch's own screen having gone to sleep, ordinary Apple Watch behaviour rather than anything MDM has lost. Tap the Watch screen to wake it, and the connection returns.

Installing the Watch app and getting it paired is a short, one-time task, covered step by step in the Apple Watch Companion Quick Start Guide.

3. The badge, and the trap in it

While a session is running, a BG badge sits beside the MIDI Health badge — and the two are answering completely different questions. The BG badge reports exactly one thing: whether the session itself is still running. It says nothing about your equipment, and was never meant to. The MIDI Health badge, entirely separate, reports connection health, on screen or off. In the foreground you get two single-purpose badges. In the background, covered in section 3, both questions are answered through one surface instead — not because either badge learns the other's job, but because a badge is not the surface available there.

The badge is not the thing to watch for.

The lock-screen prompt and the Dynamic Island are.

A prompt may appear on the lock screen after about two hours. Accepting it keeps the session alive. MIDI keeps flowing while it is shown, so you lose nothing by finishing the song first. This is the moment worth watching for — it does more for a long set than anything the badge shows.

MDM also tries to restart a session on its own. If a session stops, MDM attempts to bring it back automatically — up to three additional tries — whether or not you saw or answered a prompt. There is nothing to tap for this; it happens on its own, or it does not, and either way you find out by trying to play.

If the badge does show a colour, amber means starting or restarting, never ended. A session that has genuinely ended, with nothing left to try, shows red instead — never amber. Sessions have been observed running for many hours in practice, well past any single figure worth quoting here, which is why the prompt and the automatic restarts matter more than watching a clock.

For a long service, a wedding, or a full night with two bands, stop the session during a break and start another anyway. The automatic attempts and the prompt are a safety net, not a reason to stop planning around a break — the clock begins again, and you choose the moment rather than having it chosen for you.

Red is not what you see when a session ends. Stopping one yourself never produces it, either way you do it. Stop it from inside MDM and the badge simply clears — no colour, nothing to read. Stop it from the Dynamic Island and MDM closes entirely; there is no app left running to show any badge at all, which can look like the badge vanished, but nothing was ever red. Red means something rarer and more specific: the system has tried and failed to hold a session you never asked to end, repeatedly, until MDM's own automatic attempts to bring it back were exhausted. If you see red, start a fresh session; if it returns immediately, MDM will not be able to hold a session on that device and Split View on an iPad is the reliable alternative.

No badge means no session, which is normal. It is the ordinary state for every musician playing in Split View, and for everyone who has not started one. An absent badge has never meant anything is wrong.

4. The indicator in the Dynamic Island

While a session runs, iOS shows it in the Dynamic Island from whatever app you are in. In the foreground, whether the session is running and whether your equipment is still connected are two separate badges. Off screen, the island is the only surface there is, so it answers both questions at once rather than one badge quietly taking over the other's job. Since Build 108, if a piece of equipment drops while a session is running, the island shows an orange triangle with an exclamation mark and names the equipment directly, and the first time that happens MDM sends one silent notification alongside it. Neither needs you to look — both are simply there if you do.

Three of the things you can do with it look similar and do very different things.

- **Swiping it away hides it only.** The session keeps running. This is safe, and useful if the indicator is in your way.
- **Stopping the session from the indicator ends it — and iOS then closes MDM.** You come back to a cold start, with MIDI and Bluetooth reconnecting from nothing. Avoid this between songs.
- **Stopping it from inside MDM ends the session and leaves the app running.** This is the one to use.
- **Tapping the indicator expands it; tapping again takes you straight back to MDM.** The quickest way back to your Pads without hunting for the app.

The distinction that costs you a song is the second one. Ending a session from the iOS indicator and ending it inside MDM look like the same action and are not. One leaves you ready to play; the other leaves you waiting for Bluetooth.

5. When something stops answering

MDM watches your rig while you play, and it does so in two separate ways. Both live in **Tools → Settings**, both are set to thirty seconds, and they are easy to confuse — but they answer different questions.

MIDI Health Monitor watches the connections. It is what notices that an adaptor has gone, and it is what turns the MIDI Health badge orange when one does.

The badge is also the fastest way to act. It sits at the top right of every screen, just below the title bar — and while a background session is running, it is the right-hand of the two capsules there. Tapping it opens **MIDI Health**, which shows the state of every connection and carries both recovery actions at the foot. If anything in your rig is on a Bluetooth adaptor, **Reconnect BLE MIDI** comes first — refreshing MIDI connections cannot help while the adaptor itself is gone. **Refresh Device MIDI Connections** then reconnects ports without waiting for the next check. The label briefly confirms itself — Refreshed, with a checkmark — before returning to normal, so a tap in a hurry does not leave you wondering whether it registered.

The badge appears only when the MIDI Health Monitor is switched on. If you have turned it off, the actions are still there — **Reconnect BLE MIDI** on the Tools page, and **Refresh Device MIDI Connections** in Tools → Settings — but nothing will be watching, and nothing will turn orange when a connection drops.

MIDI Device Confirmation watches identities. Its **Auto Check** setting is how often MDM asks each device to confirm that it is still the equipment MDM thinks it is — and it also notices when something that used to answer has stopped answering at all.

In short: the Health Monitor tells you the road has gone; Auto Check tells you the thing at the far end is not who it claimed to be. The dot and the status line always move together — a connection lost and an identity that does not reconcile both show orange on both at once, never one without the other. What tells them apart is the line's own words: “Assigned MIDI port unavailable” for a connection gone, a different message when it is identity that has not reconciled. Part Two, section 4 explains how to read it.

Thirty seconds is the default on both, and it is a compromise. You can change either. Checking more often notices a loss sooner; it also means more MIDI traffic competing with the notes you are playing, which matters most on Bluetooth. Thirty seconds still notices a loss well inside a song. Five seconds is worth using while you are working out a connection problem at home, and is not worth using on stage.

Manual turns Auto Check off completely. There is no symptom until something drops and nothing tells you. It exists for musicians who want it, and MDM says as much when you select it — but on stage, leave it on an interval.

And there is a gap worth knowing about. MDM keeps watching connections even while a Background Session is running and it is off screen, trying again on a slower cadence — so a Bluetooth adaptor that reconnects on its own usually confirms itself without you doing anything. What it cannot do while off screen is show you that this happened, or let you act sooner. Bring MDM to the front and you see the current state and can use the recovery actions directly, rather than wait for the next automatic check.

Which makes one habit worth having: if a piece of equipment goes quiet during a session and everything else is still responding, bring MDM to the front before investigating anything else. That single action fixes the most likely cause. If it does not, tap the MIDI Health badge and use the actions there — quicker than anything else you could do.

6. Ten minutes before you play

A failure discovered at home is a puzzle. The same failure discovered while the band waits is a different problem, and the answer is not the same.

Recover the performance first. Diagnose the cause later.

Appendix B carries the sequence. It is the emergency card, and it is the one to follow — seven steps, in the order that gets you playing rather than the order that finds the cause. What follows here is why that order is what it is.

- **Why one Pad first.** Tapping something you know works tells you how much is wrong. Several devices silent at once points at one connection rather than several faults.
- **Why MDM comes to the front before anything else.** If a Background Session was running, MDM kept watching even off screen — bringing it forward now shows you what it already knows rather than starting the checking from nothing. If no session was running, MDM was fully suspended and has noticed nothing, and coming to the front is what starts checking again. Either way, it is also a precondition of the next step, since the recovery actions are inside MDM.
- **Why the status line before the dot.** A missing port makes everything downstream of it meaningless. No amount of checking identity or channel helps while the road is gone.
- **Why the adaptor rather than the equipment.** A Bluetooth adaptor is the likeliest thing in the chain to have let go, and it restarts in seconds where your equipment does not.
- **Why playing without it is a real option.** A Pad entered by hand changes patch on almost anything and takes under a minute. One song with a patch set by hand beats five minutes spent fixing an index.

What not to do with ten minutes to go: do not start rebuilding Devices, do not rescan MIDI Studio, and do not change settings you will not remember afterwards. Those are for tomorrow, with the equipment in front of you and nobody waiting.

7. What to take from this part

- Split View keeps MIDI running on iPad with no session, no badge and no limit.
- A Background Session works on iOS and iPadOS 26 and later. On iPhone it is the only route; on iPad it is a choice, and many prefer it.
- **Amber means starting or restarting, never ended.** A session that has genuinely ended shows red instead.
- A lock-screen prompt may appear after about two hours; accepting it keeps a session going far longer. MDM also tries restarting a stopped session automatically, up to three times, whether or not you see a prompt.
- No badge means no session, which is normal and not a fault. Red is the only badge state reporting a real failure.
- End a session from inside MDM, never from the iOS indicator — the indicator route closes the app.
- If something goes quiet during a session, bring MDM to the front first.
- Two watchers, both set in Tools → Settings: the MIDI Health Monitor watches connections, Auto Check watches identities.
- Leave both on an interval when you are performing. Five seconds is for solving a problem at home.

- The MIDI Health badge is the quickest recovery when something has dropped — one tap from any screen, and both actions are there.
- Ten minutes before you play: recover the performance, and leave the diagnosis for tomorrow.

Part Six

Reference

The MIDI facts behind everything in the previous five parts.

How to use this part

This part is for looking things up, not for reading through. It explains the MIDI messages MDM uses, what each one does, and where each turns up in the application.

You do not need any of it to use MDM. You will want some of it the first time something behaves oddly, and most of it if you ever set out to support a piece of equipment MDM has never seen.

Nothing here is specific to MIDIDeviceManager unless it says so. MIDI has been in use since 1983 and these definitions apply to any equipment that speaks it.

1. MIDI channels, and the number that is not a channel

A MIDI cable carries sixteen channels at once. Every ordinary message — a patch change, a control movement, a note — is addressed to one of them, and equipment listening on a different channel ignores it completely.

That is how several pieces of equipment share one connection. Each listens on its own channel and hears only what is meant for it.

Two numbers, and they are not the same number

Setting one when you meant the other is the commonest reason a Device looks perfect and does nothing.

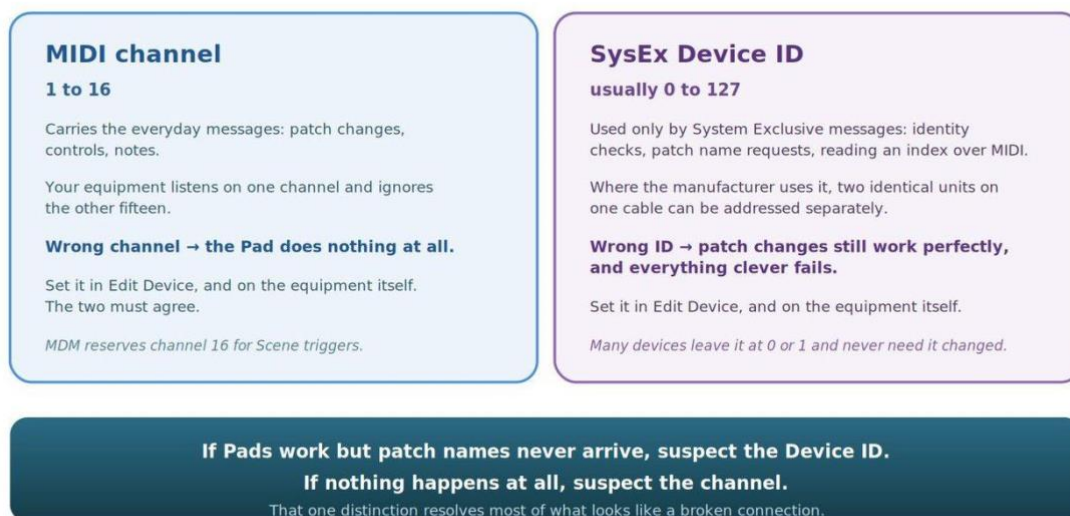


Figure 6.1 — The channel and the Device ID answer different questions.

A wrong channel is silent, not broken. Nothing is reported, no error appears, and the Device may well show green — because MDM can still reach the equipment, and reaching is a different question from being listened to.

The SysEx Device ID is a separate number entirely. It appears in System Exclusive messages only, which means identity checks, patch name requests and reading an index from the equipment. Where a manufacturer's protocol uses it, two identical units on one cable can be told apart by it. Setting it wrongly leaves patch changes working perfectly while every SysEx feature fails quietly — which is why the two are worth keeping straight.

Channel 16 belongs to MDM. It listens for Scene triggers there, and never confuses this with a Device — each Device's connection is handled independently, and MDM never relays or forwards a trigger message to any Device's output. But if your own MIDI setup routes a trigger source's signal into the same path as a Device set to channel 16 — a MIDI THRU box, a merge box, or several instruments daisy-chained on one MIDI cable — that Device would receive and respond to the same message, entirely outside MDM's control. Leaving channel 16 free means you never have to know whether your own rig works that way.

Not everything defaults to channel 1. Some equipment ships on channel 10 or elsewhere, and a manual stating a default is not evidence of what your unit is set to today. Read the channel from the equipment's own menu.

2. Program Change

A Program Change tells a piece of equipment to select a patch. It is the message MDM sends most, and virtually everything with a MIDI socket understands it.

Values run from 0 to 127, which addresses 128 patches.

MIDI counts from 0. Most equipment displays from 1.

Patch 1 on the front panel is Program Change 0.

That offset accounts for a good proportion of the confusion around Program Change, and it is worth checking whenever a Pad recalls the patch immediately before or after the one you wanted.

Patch shown on the equipment	Program Change sent
1	0
10	9
50	49
128	127

Not every manufacturer follows the same convention, and a few number their displays from 0 as well. If a Pad is consistently one patch out, this is the reason.

3. Bank Select

128 patches is not many, and most modern equipment holds far more. Bank Select is how the rest are reached: it chooses a bank, and the Program Change that follows chooses a patch within it.

Bank Select comes in two parts. MSB, the Most Significant Byte, is Control Change 0. LSB, the Least Significant Byte, is Control Change 32. Some equipment uses only MSB and some needs both; equipment that ignores LSB altogether is documented, though we have not met any ourselves.

When both are used, the order is fixed:

- Bank Select MSB (CC 0)
- Bank Select LSB (CC 32)
- Program Change

Selecting a patch above 128 is not one message. It is two or three, sent together, and they only work in that order.

This matters more than it looks. Anything reading incoming MIDI has to handle the whole group, not just the first message in it — and equipment announcing its own patch change sends the same pair. MDM stores MSB and LSB independently for every Pad, so equipment with unusual bank arrangements can be addressed correctly.

4. Control Change

Where a Program Change selects a patch, a Control Change adjusts something while the patch stays where it is. Volume, expression, modulation depth, delay mix, tap tempo, an effect switched in or out.

Values run from 0 to 127. On your equipment, some controls read that as a position — a pedal part-way down — while others are documented as treating anything above 63 as on and anything below as off. That convention belongs to the equipment, not to MDM.

Function	Common Control Change number
Modulation	CC 1
Volume	CC 7
Expression	CC 11
Bank Select MSB	CC 0
Bank Select LSB	CC 32

An expression pedal sends a stream of them as it moves, one after another, following the pedal's position.

In MDM, Control Change works both ways. A Pad's main job is a Program Change, and it can optionally carry one fixed Control Change alongside it — set once, sent the same way every tap. That is Control Change going out. Coming in, it fires Pads and Scenes: a footswitch or a backing track sends one, MDM recognises it, and the Pad it is attached to does its work. That is what the **Triggered by** settings choose. Sliders and Buttons use Control Change outgoing too, continuously or immediately rather than tied to a patch — Part Four covers them.

MDM matches the exact value it learned. A trigger learned at 127 fires on 127 and on nothing else — there is no threshold and no range. This is why learning a trigger by pressing the pedal is not merely convenient but the only reliable route: type a value you believe your controller sends, get it wrong by one, and the Pad will never fire.

5. Identity Request

An Identity Request is a standard message asking a piece of equipment to say what it is. Equipment that supports it replies with its manufacturer, its family and its model.

It is the message behind Scan MIDI Studio, behind the green and orange indicators, and behind everything in Part One.

Answering is optional and always has been. A great deal of equipment — including current, professional, expensive equipment — does not implement it, and works perfectly in every other respect. An unanswered Identity Request means MDM does not know what something is, not that anything is wrong.

The reply is addressed by SysEx Device ID, not by MIDI channel. An identity check that never comes back while patch changes work is a Device ID mismatch far more often than a connection problem.

6. System Exclusive

System Exclusive — SysEx — is the part of MIDI each manufacturer defines for itself. Where Program Change and Control Change mean the same thing to everyone, a SysEx message means whatever its manufacturer decided.

Every SysEx message begins with **F0** and ends with **F7**. Everything between is the manufacturer's own arrangement, and typically contains:

- a manufacturer identifier
- a device identifier
- a model identifier
- a command
- data
- sometimes a checksum, so the receiver can tell whether the message arrived intact

Manufacturer identifiers are assigned centrally by the MIDI Association. Older manufacturers hold a single byte; later ones hold three. The name on the register is the company that applied for it, which is not always the brand on the box — Strymon's identifier is registered to Damage Control Engineering, the company's original name, and TC-Helicon and TC Electronic hold separate identifiers despite the shared name. MDM shows the familiar brand where it can.

Registered holder	Identifier
Roland Corporation — also Boss and Ediol	41
Line 6	00 01 0C
Damage Control Engineering — Strymon's registered name	00 01 55
TC-Helicon Vocal Technologies	00 01 38
TC Electronic — a separate holder	00 20 1F

SysEx is what makes patch names possible. Program Change can recall patch 47; asking your equipment what patch 47 is called takes SysEx. Index imports, current patch requests and identity checks all travel this way. A name can of course reach an index by other routes — a file from your equipment's own software, a spreadsheet you fill in yourself — but asking the equipment directly is SysEx work.

Most musicians never need to know any of this. It matters when you set out to support equipment MDM has not met, which is what the SysEx Lab is for.

7. Glossary

Term	Meaning
Bank Select	A pair of Control Change messages, CC 0 and CC 32, choosing a bank before a Program Change chooses a patch within it.
Button	Always created as an On/Off Pair, in Edit Device. Each half sends its own fixed Control Change value when tapped; nothing is tracked, and nothing alternates.
Class-compliant	Equipment that can describe itself to an iPad or iPhone without a driver. Equipment that is not class-compliant has a USB port that will never appear.
Control Change (CC)	A message that adjusts something without changing patch. In MDM, triggers Pads and Scenes coming in; sent outgoing by a Pad's optional Control Change, and by Sliders and Buttons.
Device	The entry you create inside MDM to represent a piece of your equipment. Not the equipment itself.
Device ID	A number carried in SysEx messages. Where a manufacturer's protocol uses it, identical units on one cable can be addressed separately. Not the MIDI channel.
Equipment	Anything of yours that speaks MIDI: a synthesizer, a pedal, a rack unit, a lighting controller, another app.
Identity Request	A standard message asking equipment what it is. Answering is optional.
MIDI channel	One of sixteen addresses on a MIDI connection. Equipment listens on one and ignores the rest.
Pad	One patch change, sent to one Device, when you tap it — plus one fixed Control Change if you have chosen to add one.
Patch	An addressable, recallable location on your equipment. Some manufacturers call it a preset, a program, a rig or a scene.
Program Change	A message selecting a patch. Values 0 to 127.
Scene	One tap that prepares several Devices at once.
Searchable Instrument Index	MDM's record of where your patches live and what they are called — their addresses, not the patches themselves.
Setlist	Scenes in the order you will play them.

Slider	A Device control, created in Edit Device, sending Control Change across 0 to 127. Separate from any Pad, and stays in effect after a Pad changes the patch.
SysEx	System Exclusive. The part of MIDI each manufacturer defines for itself. Begins F0, ends F7.
USB Client	The side of a USB connection that waits to be addressed. An iPad or iPhone is a Client until something — usually the Camera Adapter — puts it into Host role instead. Not to be confused with a Device, MDM's own term for the entry representing a piece of equipment.
USB Host	The side of a USB connection in charge of it. Every USB connection needs exactly one. The Camera Adapter puts your iPad or iPhone into this role, which is what lets a class-compliant piece of equipment be recognised at all.

Appendix A — Where to look when something is wrong

Find your symptom. The Manual explains the cause; the Quick Start Guides carry the steps.

If you are about to play and something has failed, go straight to Appendix B instead.

Nothing is reaching my equipment

What you are seeing	Where to look
I tap a Pad and nothing happens at all	Part Two, section 5 — four checks, in order. The MIDI channel is the likeliest answer.
The Device is green but silent	Part Two, section 5. Green means MDM can reach it, not that it is listening.
The Device says “Assigned MIDI port unavailable”	Part Two, section 4. The adaptor or cable is gone. Nothing else matters until it is back.
It worked yesterday and does not today	Part Two, section 4 — read the status line before the coloured dot.
Only one piece of equipment is silent; the rest are fine	Part Two, section 5. Check that one's channel, then its own menus.
Everything went silent at once	Part Five, section 6. Suspect one connection rather than several faults.

MDM cannot find my equipment

What you are seeing	Where to look
A scan finds nothing	Part Two, section 2. Usually means your equipment does not answer identity requests, which is normal.
A scan finds fewer things than I have connected	Part One, section 4. Only equipment that answers can be found.
A scan offers something that is not an instrument	Part One, section 4 — interfaces and adaptors answer for themselves. Leave them switched off.
No port appears at all when I connect by USB	Part Two, section 1. Suspect the cable, then whether the equipment is class-compliant.
I cabled two pieces of equipment together by USB and neither sees the other	Part Two, section 1. Two USB Clients with no Host between them will never talk, regardless of the cable.
A Device turned orange after I rewired	Part Two, section 4 — read the status line first. If it is an identity mismatch, Part One, section 8.
“MIDI Identity not confirmed”, permanently	Part One, section 7. For a great deal of equipment this is simply true, and harmless.

Patches and indexes

What you are seeing	Where to look
Patches are missing after an import	Part Three, section 2. Build the index again over a cable.
The import finished but the names are wrong	Part Three, section 1 if you used a factory list — it describes the equipment as it left the factory. If you read the patch list from the equipment itself, section 2: a transfer can complete and still return names out of step.
I do not know which import route applies to me	Part Three, section 1. Work down the four and take the first that fits.
MDM does not follow when I change patch on the front panel	Part Three, section 5. Your equipment must transmit Program Change, and the Passive MIDI Monitor probe must have been run once.
Patch names never arrive, but Pads work	Part Six, section 1 — suspect the SysEx Device ID rather than the channel. Part Three, section 5 if your equipment has no current-patch request configured at all.
A Pad recalls the patch next to the one I wanted	Part Six, section 2. MIDI counts from 0; most equipment displays from 1.

While you are playing

What you are seeing	Where to look
MIDI stops when I switch to another app	Part Five, section 1. iPad: Split View. iPhone: a Background Session, iOS 26 and later.
The BG badge has turned amber	Part Five, section 3. Means starting or restarting, never ended — stopping a session yourself shows no colour at all, not red.
The BG badge is not there at all	Part Five, section 3. No badge means no session, which is normal and not a fault.
The BG badge has turned red	Part Five, section 3. Not from stopping a session yourself — that never shows red. Means repeated automatic-restart failure on a session you never asked to end.
Something went quiet mid-set	Part Five, section 5. Bring MDM to the front first, then tap the MIDI Health badge.
MDM closed itself when I stopped the session	Part Five, section 4. End a session from inside MDM, not from the iOS indicator.
My footswitch does nothing	Part Four, section 7. Learn the trigger by pressing the pedal rather than looking up what it sends.

Still stuck

First, rule MDM out. Try the equipment from another MIDI source — a keyboard, a foot controller, another app. If it ignores those too, the problem is not MDM and never was, and everything below is beside the point.

Copy Diagnostics, in **Tools** → **Settings**, gathers technical information about your MIDI connections and Devices — not your indexes, Pads, Scenes or Setlists. Paste it into a message and [send it to us](#). It answers most of the questions we would otherwise have to ask you.

And if your equipment is doing nothing at all and you cannot find the cause, [tell us what you have](#) — some equipment ships with a setting that quietly discards incoming MIDI.

Appendix B — Ten minutes before you play

This appendix is the emergency procedure. It is the one to follow, and Part Five explains why the order is what it is rather than repeating it. The same seven steps are published as the MDM Live Emergency Guide, a one-page card for printing and keeping with your equipment.

Recover the performance first.

Diagnose the cause later.

A failure at home is a puzzle. The same failure with the band waiting is a different problem, and the answer is not the same. Work in this order. It is not the order that finds the cause — it is the order that gets you playing.

1. Find out how much is wrong

Tap a Pad you know works. If several pieces of equipment are silent at once, suspect one connection rather than several faults — an adaptor, a cable, a hub.

2. Bring MDM to the front

If you started a Background Session, MDM kept watching even off screen — check the Dynamic Island or your notifications first, since it may already have named what dropped. If you didn't, MDM was fully suspended and noticed nothing. Either way, bring it forward now: the recovery actions are inside MDM, which is why this step comes before the next one.

3. Tap the MIDI Health badge

Top right of every screen, just below the title bar — and if a background session is running, it is the right-hand of the two capsules there. It turns orange within seconds of a connection dropping if MDM can still reach your equipment another way, red if it can't — either way, the actions below are your next move. Tapping it opens **MIDI Health**, with two actions at the foot. If anything in your rig is on a Bluetooth adaptor, tap **Reconnect BLE MIDI** first; refreshing MIDI connections will not help while the adaptor itself is gone. Then **Refresh Device MIDI Connections**. Both briefly confirm themselves with a checkmark before returning to normal, so you know a tap registered without waiting to see if it worked. All of this is inside MDM, which is why step 2 comes first.

If there is no badge, the MIDI Health Monitor is switched off. Both actions are still available: **Reconnect BLE MIDI** on the Tools page, and **Refresh Device MIDI Connections** in Tools → Settings.

4. Read the status line, not the colour

On the Devices screen, the line under each Device says more than the dot beside it. “Assigned MIDI port unavailable” means the adaptor or cable is gone — nothing else will help until it is back. Inside MIDI Health, the same Device reads “Assigned output not found; defaulting to [name]” if orange — still sending, just not where you expect — or “Assigned output not found and no global output is available” if red — nothing going out at all. A reconnected port usually goes green on its own; the refresh above simply gets you there sooner.

5. Power-cycle the adaptor, not the equipment

A Bluetooth adaptor is the likeliest thing in the chain to have let go, and it restarts in seconds. Your equipment takes longer and is rarely the problem.

6. Check the channel on the one that is wrong

If one Device is silent and the rest are fine, its MIDI channel is the likeliest cause. Read the channel from the equipment's own menu and match it in Edit Device.

7. Play without it

If none of the above has worked, add a Pad by hand with the Program Change number you need. It takes under a minute and works on almost anything. One song with a patch changed by hand is better than five minutes spent fixing an index.

What not to do with ten minutes to go

- Do not rebuild Devices.
- Do not rescan MIDI Studio.
- Do not rebuild an index.
- Do not change settings you will not remember afterwards.

All of those are for tomorrow, with the equipment in front of you and nobody waiting. Write down what happened while it is fresh — that is worth more than anything you could change now.

Afterwards

Copy Diagnostics in **Tools** → **Settings** captures the state of your connections. Copy it while the problem is still present if you can; it is far more useful than a description written the next day.