LeadMuse

A cross-DAW MIDI-effect plugin that plays lead and solo guitar in the style of the greats — locked to your transport.

User Manual · Version 1.1.1

VST3 · Audio Unit | Windows & macOS

© Simply Bass Productions

Welcome

If you've never used a plugin quite like this one, that's fine — LeadMuse is a fairly new kind of tool, and this manual is built to get you playing before it gets you reading theory. Chapter 1 tells you what LeadMuse actually is and how it fits into a session; Chapter 4 gets sound out of it in three steps. Everything after that is there for when you want more control, or when something isn't behaving the way you expect.

You don't need to read this front to back. Treat it the way you'd treat a good reference book: jump to the chapter that answers the question you actually have right now, and trust that the cross-references will catch you if you need something from earlier.

Contents

1 What LeadMuse is
2 Installing
3 Activating & the trial
4 Quick start (3 steps)
5 Your first song
6 The interface at a glance
7 Source modes
8 Building a song
9 The chord selector
10 Styles & the feel knobs
11 Role, licks & expression
12 Drag-MIDI — export & import
13 Sharing: styles, songs, bundles
14 Using LeadMuse in your DAW
15 Pro Tools
16 Troubleshooting
17 Appendix & reference

Figures

3.1 The License button and its states
3.2 The activation screen
5.1 Example-VerseChorus loaded in the arranger
5.2 The Verse and its two-chord bar
7.1 The Source dropdown
8.1 A section band and its bar grid
8.2 The quick song builder
8.3 Editing one bar's feel
9.1 The chord selector
10.1 The feel knobs
12.1 The drag strips
12.2 The drop-target menu
12.3 The clip-fit prompt
12.4 The replace confirmation
14.1 The Preview control

1 What LeadMuse is

LeadMuse is a MIDI-effect plugin. If that phrase is new to you, here's the short version: it doesn't make a sound by itself. Instead, it reads MIDI in and writes MIDI out, the same way a session player reads a chord chart and plays a part — except the part LeadMuse writes goes straight into whatever instrument you point it at. In every host, on every platform, the job is the same:

```
chords  →  LeadMuse  →  an instrument
```

You give LeadMuse a chord progression, pick a style, and it generates a musical part that stays locked to your host's tempo and transport. Press play in your DAW, and LeadMuse plays with you — stop, and it stops. There's no syncing to do, because it's already listening to the same clock your DAW is.

I think that's the detail worth sitting with for a second: you're not recording a performance and hoping it lines up. LeadMuse generates its part in real time against your transport, every time, so it's always in time by construction.

Two ways to drive it

LeadMuse can build a part from a progression you write ahead of time, or from chords you feed it as you play. Both roads lead to the same engine underneath.

- **Song mode** — you build a full arrangement inside LeadMuse: sections, meters, and chords per bar, laid out in its own grid. LeadMuse plays that arrangement back, start to finish, in time with your transport.
- **Follow mode** — LeadMuse listens to chords arriving at its MIDI input in real time and plays along with whatever you feed it. Hold a chord on a keyboard, or route in a MIDI clip or chord track, and LeadMuse follows.

> **More Info.** There's a third mode too — a built-in demo progression for quickly auditioning a style before you've built anything. Chapter 7 covers all three source modes in detail; for now, know that Song mode is where you'll do most of your work.

A deterministic engine

"Deterministic" is a five-dollar word for a simple promise: the same inputs always give you the same output. Feed LeadMuse the same progression, the same style, and the same **Seed** value, and you'll get exactly the same notes back, every single time.

This matters more than it might sound like it should. A generated part is only useful if you can trust it to stay put — you don't want it quietly changing on you between sessions, or between an early mix and the final bounce. Because LeadMuse is deterministic, a part you like is a part you can keep. And when you want something different, you don't need to fight the engine for it — you just nudge the **Seed** control and roll a fresh variation on demand.

What's in the box

LeadMuse comes with a fairly complete toolkit for building a part and getting it out into your session:

- 10 built-in lead-guitar styles, each with its own register, phrasing, and bend & vibrato character — not transcriptions of any real player, but distinct characters you can pick between.
- A section-based arranger with a sub-bar chord grid, plus a circle-of-fifths chord selector for building chords by ear and by eye rather than by typing symbols (Chapter 9).
- Live chord-follow, so LeadMuse can play under chords you feed it instead of ones you've written into the grid.
- Drag-out MIDI export, for pulling a finished part straight onto a track in your DAW with no routing at all (Chapter 12).
- Plain-text, shareable style, song, and bundle files, so a part — or an entire arrangement — travels easily between projects and between people (Chapter 13).

That's the whole shape of the plugin in outline. The rest of this manual walks through each piece in turn, starting with getting LeadMuse installed and into your DAW.

2 Installing

Installing LeadMuse is a normal plugin install — nothing about it is unusual, and you'll be done in a couple of minutes. The one thing worth knowing going in is *where* it lands: LeadMuse installs into the shared system folder every DAW scans for plugins, not into some private folder of its own. That's what makes "install once, use in any DAW" actually true.

I'll walk through Windows and macOS separately, since the installers — and the folders they use — are different on each.

Installing on Windows

1. Run `LeadMuse-1.1.1-Setup.exe`.
2. Accept the license agreement when the installer asks, then continue.
3. Let the installer finish. It copies the VST3 into `C:\Program Files\Common Files\VST3\LeadMuse.vst3` — the shared folder every Windows DAW scans on startup — along with the user manual, a couple of reference documents, and a folder of example songs and styles.
4. Launch your DAW. If LeadMuse doesn't show up right away, rescan plug-ins — most DAWs have a menu option for this, and a fresh install is exactly when you need it.

Notice that step 3 doesn't ask you where to install anything. That's deliberate: LeadMuse always goes to the one folder every DAW already knows to look in, so there's no path to get wrong and no separate step to point your DAW at it afterward.

> **Note.** The installer needs administrator rights, since it's writing into `Program Files` and a folder shared by every application on the system. Windows will prompt you for that automatically — you don't need to do anything differently.

Installing on macOS

1. Open `LeadMuse-1.1.1.pkg`.
2. Accept the license agreement and follow the installer through to the end.
3. The installer asks for your password, since — like the Windows install — it's writing into a folder shared system-wide rather than just your own user account. Once it's done, both plugin formats are in place:

- **VST3** → `/Library/Audio/Plug-Ins/VST3/LeadMuse.vst3` - **Audio Unit** → `/Library/Audio/Plug-Ins/Components/LeadMuse.component`

1. Launch your DAW and rescan plug-ins if LeadMuse doesn't appear on its own.

Here's the one Mac-specific decision you need to make: which format to use. **Logic Pro and UAD Luna only load the Audio Unit** — there's no VST3 option in either host — so the AU build is the one that matters there. Most other Mac DAWs are happy with either format, and it won't matter which one loads.

About security warnings

LeadMuse's Windows and macOS builds are code-signed — Windows Authenticode on the `.exe` and the plugin itself, Apple Developer ID plus notarization on the Mac side — so you shouldn't need to fight your way past an "unknown publisher" block to install it.

That said, don't be surprised if a brand-new release still draws a raised eyebrow from Windows SmartScreen or an antivirus scanner the first few days after it ships. That's a reputation check, not a signature check: SmartScreen and some antivirus engines flag *any* installer from a small, unfamiliar publisher until enough people have safely run it, signed or not. If you see a warning like that, it should clear on its own within a few days as the new release builds up a track record — it isn't a sign anything's actually wrong with the file.

> **More Info.** If your DAW can't find LeadMuse after installing, or you're chasing a routing problem rather than an install problem, Chapter 16 has a troubleshooting table, and Chapter 14 covers per-DAW routing in depth.

Formats by DAW

Format	Hosts
VST3	Reaper, Bitwig, Cubase, Studio One, FL Studio, Cakewalk Sonar, Mixbus, Digital Performer, Fender Studio Pro
Audio Unit	Logic Pro, UAD Luna, and other AU hosts (macOS)
CLAP	Reaper, Bitwig, Studio One, FL Studio, and other CLAP hosts (Windows & macOS)
AAX	Not yet — see Chapter 15 for how to use LeadMuse in Pro Tools today

An AAX build for Pro Tools is on the roadmap, but it depends on a separate signing program through Avid that's outside LeadMuse's own release process — so there's no date to promise yet. Chapter 15 covers the two ways to use LeadMuse in Pro Tools in the meantime.

Where your custom styles and songs live

The installer doesn't just place the plugin — it also drops a folder of example songs and styles alongside the documentation, so you've got something to load and pull apart on your very first session.

> **More Info.** Chapter 13 covers the full file-sharing picture: where LeadMuse looks for your own styles and songs, what the three file types (`.lmstyle` , `.lmsong` , `.lmbundle`) are for, and how to share a song with someone who doesn't have your custom styles. For now, just know the examples are there waiting for you once LeadMuse is up and running.

3 Activating & the trial

Let me get this one out of the way early, because it's the first wall a new user tends to hit — usually at the moment they first try to save. LeadMuse lets you build a whole arrangement and play it back without entering anything at all. What it holds back until you've activated is *getting your work out*: saving a song, exporting or dragging out MIDI, and the two account-connected buttons, Community and Report. So you can explore freely, but the day you want to keep something, this is the step waiting for you.

Here's the good news up front: activation is a one-time paste of a short code, and there's a free 30-day trial that unlocks *everything* — it isn't a crippled preview. A trial license and a paid license behave identically while they're valid; the only difference is that the trial has a clock on it.

What works before and after

It's worth being precise about where the line falls, because it's narrower than people expect:

- **Always available, no code needed** — building an arrangement, entering chords, choosing styles, turning the feel knobs, and pressing Play. The whole creative surface works unactivated, so you can decide whether you like LeadMuse before committing to anything.
- **Unlocked by a trial or purchase code** — **saving** and loading your own songs, **exporting and dragging out** MIDI, and the **Community** and **Report** buttons. In other words, everything that turns a good session into a file you can keep or share.

> **Note.** The 30-day trial is fully featured — save, export, drag-out and all. "Trial" here means *time*-limited, not *feature*-limited. When it ends, LeadMuse doesn't lock you out of your work; it drops back to build-and-play until you enter a purchase code.

Reading the License button

Everything happens on one small screen, and you reach it from the **License** button on LeadMuse's top row. That same button is also your status light, so it's worth knowing what it's telling you at a glance:

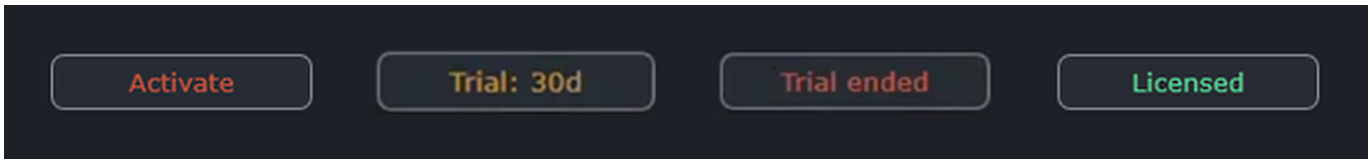

Figure 3.1 The License button doubles as your status light. From left: **Activate** (red — nothing entered yet), **Trial: Nd** (amber — a trial is running, days left shown), **Trial ended** (red — your 30 days are up), and **Licensed** (green — a paid license is active).

- **Activate** (red) — nothing entered yet; you're in build-and-play mode.
- **Trial: 12d** (amber) — a trial is running, with the days remaining shown right on the button.
- **Licensed** (green) — a purchase code is active on this machine. You're done, and you won't see this screen again unless you go looking for it.
- **Trial ended** (red) — your 30 days are up; enter a purchase code to carry on.

Activating LeadMuse

1. Click the **License** button on the top row. The activation screen opens, titled *Activate LeadMuse*.
2. Paste your code into the field — it looks like `SBP-XXXX-XXXX-XXXX-XXXX`. The same field takes a trial code and a purchase code; LeadMuse works out which one you've given it.
3. Click **Activate**. It checks in with the store over the internet, so give it a second. When it succeeds you'll see "*Activated — thank you!*", and the header button changes to **Trial: 30d** or **Licensed**.

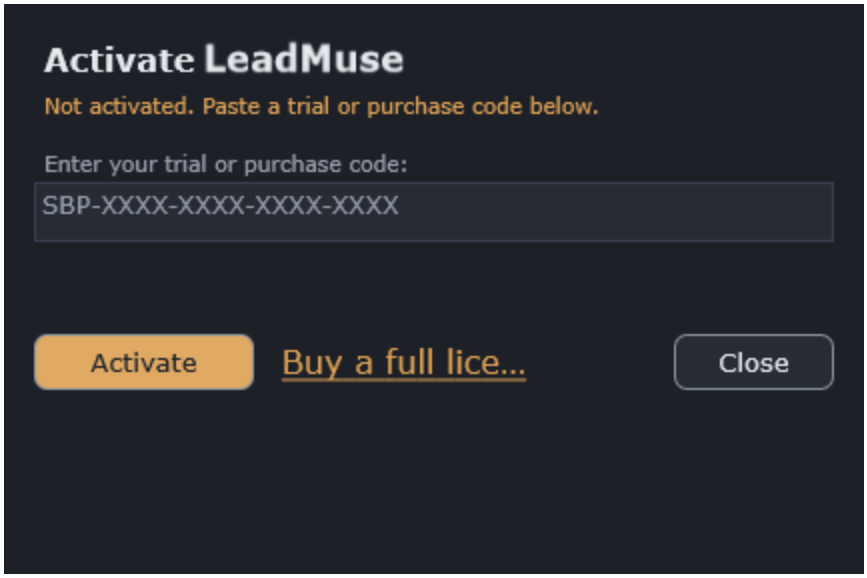

Figure 3.2 The activation screen. Paste your `SBP-...` code into the field and click **Activate**; the **Buy a full license** link goes straight to the store if you don't have a code yet. Once you're activated, a **Deactivate** button appears here too, for releasing this machine.

Notice that you only do this once per machine. After it succeeds LeadMuse remembers the license and works **offline** from then on — a paid license quietly re-checks with the store about once a month when you happen to be online, but you never have to think about it.

Where the code comes from

A **trial** code comes from the free-trial sign-up on the LeadMuse website; a **purchase** code arrives when you buy. Either way it's the same `SBP-...` format, and either way you enter it the same way. If you don't have one yet, the activation screen has a **Buy** link that takes you straight to the store.

Moving to another machine

One license covers **three machines** at a time — a studio desktop, a laptop, and one more. If you try to activate on a fourth, LeadMuse tells you the limit's reached and asks you to free one up first.

1. On a machine you're retiring, open the License screen and click **Deactivate**. That releases the seat.
2. Activate on the new machine as above.

> **Tip.** Deactivate *before* you wipe or sell a machine, not after. Once it's gone you can't click Deactivate on it — you'd have to release the seat from your account page on the website instead, which is a few more

> steps.

4 Quick start (3 steps)

Here's the fastest way to hear LeadMuse actually play — using **Song mode**, which is where you'll do most of your work anyway. Three steps, and you don't even need a progression ready to go, for a reason I'll explain in step 2.

1. **Insert LeadMuse before an instrument.** LeadMuse and the instrument need to sit in the same signal path, so LeadMuse's MIDI can reach it — exactly how depends on your DAW. In Logic, that means putting LeadMuse in the track's **MIDI FX** slot; in Reaper, it means LeadMuse and the instrument in the same FX chain, LeadMuse first. Those are just two examples to get you oriented — Chapter 14 has the full, host-by-host walkthrough.
2. **Look at the chord grid — or write your own.** Open LeadMuse's **Arrangement** grid and you'll find it isn't empty: every fresh instance of LeadMuse starts with a two-section demo already built in, chords and all. You can press Play right now and hear something. When you're ready to write your own progression, click a chord cell to open the **chord selector** — that's how every chord gets entered, by picking it visually rather than typing a symbol — and choose your chords there (Chapter 9 covers the selector in full).
3. **Pick a style and press Play.** Every section has its own style, chosen from the dropdown, and the demo sections already have one each — pick something else if you'd rather. Make sure **Source** reads **Song (grid)** (it's the default), then start your host's transport. LeadMuse plays through your instrument, locked to the beat.

Notice that nothing about this depends on you building anything first. Step 2 exists so you know your own progression is one click away, not because LeadMuse needs you to fill in a blank page before it'll make a sound.

> **Note.** No routing handy right now? **Drag-MIDI** (Chapter 12) lets you drag the finished part straight onto a track as a MIDI clip instead — no routing, no setup, works in every DAW. It does need LeadMuse activated first, though (Chapter 3) — a trial or purchase code, entered once, and you won't be asked again.

To play along live instead

Set **Source** to **Follow MIDI** and send chords into LeadMuse's MIDI input — from a MIDI clip, a keyboard, or your DAW's chord track. LeadMuse plays under whatever chords it hears, in real time, no arrangement required.

5 Your first song

The rest of this manual takes LeadMuse apart control by control. This chapter does the opposite: it walks one real song from open to finished, so you can see how the pieces fit together before you meet them one at a time. We'll load an arrangement that ships with LeadMuse, listen to it, take it apart, change it, and pull the result into your DAW — the whole loop, start to end.

There's a quiet advantage in that phrase "an arrangement that ships with LeadMuse": because the engine is deterministic, the example plays the *same notes on your machine as it does on mine*. So when I say what you'll hear, you actually will — I'm not describing a possibility, I'm describing your screen. Get your DAW open with LeadMuse on an instrument track (Chapter 4 has the three-step version if you need it), and let's go.

Step 1 — Load the example and press Play

LeadMuse's installer drops a folder of ready-made songs and styles next to the plugin. We'll open one of the songs.

1. Click **File** on LeadMuse's top row, then **Load Song**.
2. Go to `C:\Program Files\LeadMuse\Examples\Songs` (on macOS, `/Library/Application Support/LeadMuse/Examples/Songs`) and open `Example-VerseChorus.lmsong`.
3. Make sure **Source** reads **Song (grid)**, then press Play in your DAW.

LeadMuse plays the whole arrangement through your instrument, locked to your transport. You'll hear a single-note lead line move through three sections, then loop. Let it run a couple of times; the next steps are about understanding what you just heard.

Figure 5.1 `Example-VerseChorus` loaded — the Verse rolling into the Chorus, each bar's chord sitting in the grid. (The 3/4 Bridge is just off-screen to the right.)

Step 2 — Read the arrangement

Everything you're hearing is laid out in the **Arrangement** grid. Along the top are three **section bands**, each a labeled block: **Verse**, **Chorus**, and **Bridge**. A section is simply a run of bars that share a meter and a style; click a band to select it, and its settings fill the row above the grid.

Here's what's in each one:

- **Verse** — four bars of 4/4 on the **Blues Lead (B.B. Kingpin)** style, set to a **Lead** role. The chords are `C`, `Am`, then `F` moving to `G` halfway through the third bar, and back to `C`.
- **Chorus** — four bars of 4/4, stepped up to a **Solo** role on **Hard Rock (Jammy's Pages)**: `F`, `F`, `C`, `G`. Same chords, busier playing — that's the Role control, which Chapter 11 covers in full.
- **Bridge** — four bars of **3/4** this time, sparse and lyrical, on **Cello (MoMoYa)**: `Am` moving to `Dm7` partway through the first bar, then `G7`, `Cmaj7`, `E7`. Notice the meter changed and the line followed it — you didn't have to do a thing.

Under each band is the **bar grid**: numbered cells, one row per bar, where the chords live. The third Verse bar is the clearest example of the grid's reach — `F@0 G@2` means "the first chord from beat 1, the second from beat 3", and you can see both chords sitting in the one bar at different points.

Figure 5.2 The Verse's four bars. Bar 3 is the two-chord bar — `F@0 G@2` — the first chord from beat 1, the second from beat 3.

Step 3 — Make it yours

A song you can't change isn't much of a starting point, so let's change something and hear it. We'll swap a chord using the **chord selector** rather than typing — the visual picker you'll use for every chord you ever enter (Chapter 9 covers it in full).

1. Click the **Chorus** band to select it.
2. In its bar grid, click the chord in bar 2 — `F`. The chord selector opens with that chord already loaded.
3. Pick a different chord — try `Dm` — and click **Apply**. Notice how the grid cell updates and the band redraws.

4. Press Play. The chorus now moves through your new chord, and because nothing else changed, everything around it is exactly as it was.

That last part is the deterministic engine doing you a favor: you changed one chord, and *only* that chord's neighborhood changed. Nudge the **Seed** control at the top if you want a fresh take on the same chords; set it back and the old take returns, note for note.

> **Tip.** Want to hear one section on its own while you work? Tick **Selected section only** under the grid, so Play loops just the part you're editing instead of the whole song.

Step 4 — Get the part out

You've got something you like; now pull it into your session as real MIDI. This is where a trial or purchase code earns its keep, because drag-out is one of the things activation unlocks (Chapter 3).

1. Find the two drag strips under the grid: **Drag Lead to your DAW** and **Drag Chords to your DAW**.
2. Click and hold **Drag Lead to your DAW**, drag onto an empty track in your DAW, and let go.
3. LeadMuse writes the whole arrangement to a MIDI clip and drops it there — the exact part you were hearing, now editable in your DAW like any other clip.

That's the finished loop: an idea, played, shaped, and printed to a clip. (Figure 12.1, in the drag-out chapter, shows the two strips.) From here Chapter 12 covers drag-out in full — including dragging just the chords, or just the selected section.

Step 5 — Keep it

Last thing: save your version so it's yours. Click **File ▸ Save Song**, give it a name, and LeadMuse writes a `.lmsong` file — a small, plain-text arrangement you can reopen, back up, or hand to someone else (Chapter 13). Save it somewhere of your own rather than back into the read-only `Examples` folder, so the original stays put for next time.

And that's a whole first song. You loaded a real arrangement, understood its sections and chords, changed one, dragged the result into your DAW, and saved your own copy. Every control you touched has a chapter of its own ahead — but you've already seen how they add up.

6 The interface at a glance

The editor is one window, and everything in it is organized top to bottom by scope: global controls that apply to the whole song, then the settings for whichever section you've got selected, then the arrangement grid itself. Once you know that ordering, the window stops looking crowded and starts looking like exactly three questions, answered in order: *what's the song doing, what's this section doing, and what are its chords.*

Top row — global controls

These apply to the whole song, no matter which section is selected.

Control	What it does
Source	Song (grid) · Follow MIDI · Demo progression (Chapter 7)
Key	Key: Auto (detected from your chords) or force a major/minor key
Host BPM	Read-out of the host tempo LeadMuse is locked to
License / Activate	Shows your activation status — trial days remaining, or **Activate** if you haven't yet — and opens the activation screen
File	Save / Load Song, Save / Load Bundle, Build arrangement, Clear Song, Follow section on play, and a handful of smaller settings covered in later chapters (Chapters 8, 10, 12, 13)
Community	Browse and share styles, songs and bundles with other owners — requires activation
Report	Send a bug, idea or question to the feedback board (Chapter 16)
Bypass	Silences LeadMuse's output while it stays in the chain — a "mute" for the plugin. Saved with the song
Seed	Re-rolls the part; same seed = same notes
Octave	Global transpose in octaves
Range (Low / High)	Folds every generated note by octaves to stay inside your instrument's playable window
Drum	Drum-lock amount, 0→1 (Chapter 11)

The **File** menu is worth a callout of its own: it's the one control on this row that hides more than it shows. Beyond the items above, it's also where you turn on live chord capture, learn a variation-trigger note, assign a MIDI controller to the Drum knob, and a few other settings that each get their own explanation where they're actually used.

> **More Info.** Activation — the trial, what's gated, and what a purchase code unlocks — has a chapter of its own: see Chapter 3. In short, LeadMuse plays and lets you build a song fully unlicensed; a trial or purchase code is what unlocks saving, exporting and drag-out MIDI, and the Community and Report buttons.

Selected Section — per-section settings

Every section — Verse, Chorus, or whatever you name it — carries its own settings, so a 4/4 Motown verse and a 3/4 waltz chorus can genuinely be different pieces of music living in the same song.

Control	What it does
Name	Label for the section (Verse, Chorus, ...)
Meter	Time signature for this section (4/4, 3/4, 5/4, ...)
Style	Which style plays this section
Mode	The scale mode for this section's connective notes — Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian or Locrian, or **Inherit** to just follow the song key

The **feel knobs** are their own small family, and picking a style loads a full set of defaults for them — you're always adjusting from a sensible starting point, never a blank slate.

Knob	Effect
Density	How busy the part is — more notes vs. more space
Swing	Delays the off-beat for a shuffle feel
Humanize	Loosens timing and velocity so it feels played, not typed
Ring	Note length — from short and detached to long, held and connected
Bend	How often a held note bends up a whole step (pitch-bend) and settles back
Vibrato	How often a held note gets a vibrato swell (CC1)
Runs	Fast-run tendency — longer, busier licks vs. sparse, spacious phrasing

> **More Info.** Mode and the feel knobs (including Anchor, on BassMuse) all get a full explanation with audible before/after context in Chapter 10. This table is just here so you know what you're looking at.

Arrangement — the chord grid

- **Section bands** run across the top — click one to select it.
- **Chord cells** sit below the bands, one per grid tick. Click a cell to open the **chord selector** (Chapter 9) and choose a chord visually — that's the only way to set a chord; there's no way to type a symbol directly into a cell.
- **Grid** sets the tick size — `1/4`, `1/8`, or `1/16` — so you can place more than one chord per bar when a passage needs it.
- **+ Section / − Section**, **+ Bar / − Bar**, and the zoom **+ / −** buttons shape the arrangement as you build it.

A **playhead** sweeps the grid during playback and auto-scrolls to keep up, and the selected section follows it — so whichever section the song is currently playing, that section's knobs and chords are the ones on screen. It's considerate about it, too: the playhead steps aside while you're actively editing — holding a knob, typing the section name, or browsing the style or meter list — and for about a second after any change, so it never

yanks the selection out from under you mid-tweak. It's on by default; turn it off with **File → Follow section on play** if you'd rather the selection stay put.

Styles & export

- **Import Style / Export Style** — share `.lmstyle` files (Chapter 13).
- **Drag MIDI to your DAW** — press and drag to drop the whole arrangement as a `.mid` file (Chapter 12).
- **Drag Chords to your DAW** — the strip next to it does the same thing for just the chords, as sustained block-chord MIDI for a piano or pad track to comp under the part (Chapter 12).

7 Source modes

The **Source** dropdown answers one question: where should LeadMuse get its chords from? Think of it as choosing between three different musicians' jobs — reading a chart you wrote out in advance, following a singer by ear in real time, or just noodling along to a practice loop. All three play through the exact same engine underneath; they just differ in where the chords come from.

There are three settings, and picking the right one matters because two of them quietly change what the rest of the window looks like.

Figure 7.1 The **Source** dropdown — here on **Song (grid)** — with the time-signature picker beside it.

Song (grid)

This is the mode you'll use to actually write a part into a song. LeadMuse plays the arrangement you built in the grid — every section, in order, with each section's own meter, style, feel, and chords — locked to your host's transport from the first bar to the last.

Song mode is also the only one of the three that plays through *the whole arrangement*. Follow MIDI and Demo progression, by contrast, both loop — which is exactly why they're built for auditioning, not for a finished take.

Follow MIDI

Follow MIDI turns LeadMuse into a real-time accompanist instead of an arranger. It ignores the grid completely and instead watches the notes arriving at its own MIDI input, live.

Here's the mechanism, since it explains a lot about how the mode behaves:

1. **Hold a chord.** Play it on a keyboard, or route in a MIDI clip or your DAW's chord track — anything that sends note-on/note-off into LeadMuse's input works.
2. **LeadMuse figures out what chord you're holding**, using the lowest note as the root, and generates a one-bar loop of that chord using the currently selected style and feel.
3. **Change the chord, and it regenerates** — but only when the chord actually changes, so it isn't re-rolling the pattern on every note.
4. **Let go of every note, and it goes quiet.** No held notes means no chord to play under, so LeadMuse simply stops rather than holding onto the last thing you played.

> **Note.** Channel 10 is reserved for drum-lock input (Chapter 11) and is never read as a chord tone — so you can feed a kick pattern and a chord track into the same MIDI input at once without them interfering with each other.

Because there's no arrangement involved, the grid hides itself in this mode — there's nothing in it to look at. The window also shrinks down to a compact height, and the section panel below relabels itself **FOLLOW STYLE (generated on each chord you hold)**, which is LeadMuse telling you plainly what it's doing.

Demo progression

Demo progression is the fastest way to answer "what does this style actually sound like?" without building anything first. LeadMuse loops a built-in eight-chord progression through whichever style is currently selected, so you can flip through styles and hear each one immediately.

Like Follow MIDI, it loops indefinitely rather than playing once — and it hides the arrangement grid and shrinks the window the same way, since there's nothing to arrange here either.

Both Follow MIDI and Demo progression need a style and a feel to play with, and neither one has its own arrangement to pull those from. So both borrow them from the same place.

> **Tip.** Style and feel in Follow and Demo modes always come from **Section 1** of your arrangement — whatever section currently sits first in the grid. Set that section's style and feel knobs the way you want the live sound to behave, and both modes will follow. That's also why the section panel stays visible and editable even with the grid hidden: it's not decoration, it's the live control surface for whatever you're about to play or audition.

8 Building a song

A song, to LeadMuse, is a list of **sections** — Verse, Chorus, Bridge, whatever names you give them. Each section carries its own time signature, style, feel, and bars, so a 4/4 Motown verse can sit right next to a 3/4 prog bridge in a completely different style. This chapter walks through building that list up: adding and arranging sections, filling them with chords, and the handful of controls — key, seed, range, and the live variation trigger — that shape the song as a whole.

Sections

Everything starts with the row of section bands across the top of the arrangement grid.

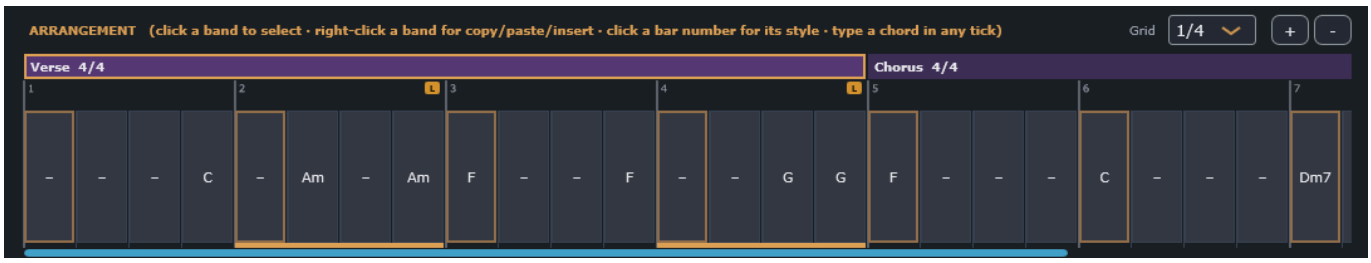

Figure 8.1 A section band and the bar grid beneath it. The band names the section, its style and meter; each numbered cell below is one bar, and clicking a cell opens the chord selector to set that bar's chord.

- Click a band to **select** it — its name, meter, style, and feel all appear in the panel below, ready to edit.
- **+ Section** adds a new one; **– Section** removes the selected one.
- A song always keeps at least one section. If you try to delete the last one, **– Section** simply flashes to let you know rather than leaving you with nothing to edit.
- To wipe the whole song and start fresh, use **File** → **Clear Song**.

The quick song builder

Building a multi-section arrangement one **+ Section** click at a time works, but it's not the fastest way to get a whole song's skeleton down. **File** → **Build arrangement...** does it in one step.

1. Open **File** → **Build arrangement...**.
2. Type your song's structure as `Name:Bars` tokens, separated by spaces or commas — for example: `Verse:8 Prechorus:4 Chorus:8 Turn:4 Verse:8 Chorus:8`. Or skip typing and click a **template** instead: **Pop**, **Verse/Chorus**, **AABA**, or **12-bar blues**.
3. Choose **Append** to add the new sections to the end of your current song, or **Replace** to clear the song first and start over.

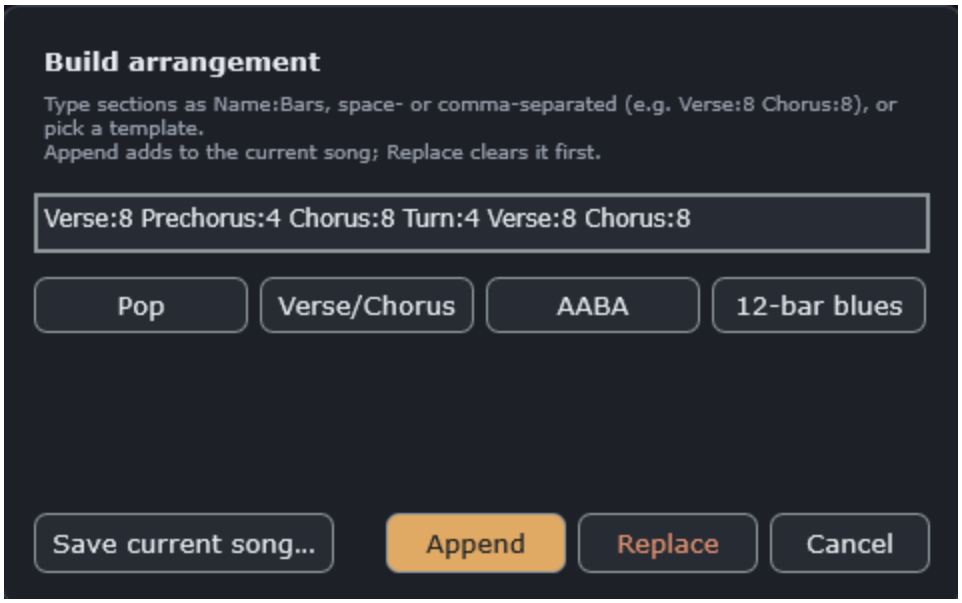

Figure 8.2 The quick song builder. Type a `Name:Bars` structure or click a template, then **Append** or **Replace**.

LeadMuse creates every named section with the bar count you gave it (clamped to a sane 1–64 bars if you type something wild), and each new section inherits the meter, style, and feel of whichever section was selected when you opened the builder — so the sections you build already carry the groove you'd already dialed in, rather than starting from scratch. Type the same name twice and LeadMuse numbers them for you automatically (*Verse 1*, *Verse 2*), so you never end up with two sections both just called "Verse."

> **Tip.** About to hit **Replace**? Click **Save current song...** first. It backs up your existing arrangement to a `.lmsong` file before Replace clears it, so a structure experiment can never cost you the song you already had.

Copy, paste & insert sections

Right-click any section band for the full set of section-level actions:

- **Insert Section Before / After** — drop a new section at that exact spot, not just at the end.
- **Duplicate Section** — copy the section (chords, meter, style, feel) in place, named "... copy".
- **Copy / Cut**, then **Paste Section Before / After** — move or repeat a section anywhere in the song (copy a chorus once, paste it everywhere it recurs).
- **Copy Chords / Paste Chords** — copy just this section's chord progression into another section, *without* touching the target's style, feel, or meter. This is the one to reach for when you want a chorus's chords in a bridge that should otherwise keep its own voice — Copy Section would carry the chorus's style and feel along with it, which usually isn't what you want.
- **Time signature ▸** — set this section's meter right from the band, without opening the section panel. Thirteen meters are on the menu: `4/4 3/4 2/4 5/4 6/4 7/4 2/2 3/2 6/8 9/8 12/8 7/8 5/8`. The odd-eighth meters (7/8, 5/8) now play at their true length — 7/8 is 3½ beats, not rounded up to 4 — so loops, the playhead, and MIDI export all stay in sync.

- **Grid resolution ▸** — **Default** (follow the global grid) or **1/4 / 1/8 / 1/16** for this section only. It's purely a view aid for placing chords precisely in a busy section; it never changes playback, and your choice is saved with the song.
- **Seed ▸, Octave ▸** — give this section its own value instead of the song-wide one, or **Inherit** to hand it back to the song (see *Seed, octave & range*, below).
- **Mute Section (silence)** — see *Silence*, below.
- **Split Section ▸ Before bar N** — divide the section in two at that bar, keeping its meter, style, and feel on both halves. Handy after a dropped MIDI clip grows a section past where you wanted the split. The bar numbers in this menu count from the start of the whole song, not from the start of the section — so "Before bar 12" means the twelfth bar of the arrangement, even if this section starts at bar 9.
- **Delete Section**.

Keyboard shortcuts do the same jobs when the editor has focus: **Ctrl/Cmd+C** copy, **Ctrl/Cmd+X** cut, **Ctrl/Cmd+V** paste after, **Ctrl/Cmd+D** duplicate.

Undo & redo

Every change you make to the arrangement can be undone with **Ctrl/Cmd+Z** and redone with **Ctrl/Cmd+Shift+Z** (on Windows, **Ctrl+Y** redoes too) — adding, deleting, reordering, renaming or duplicating sections; editing chords; changing a section's meter, key, mode, style or feel knobs; per-bar fills and mutes; and dropping a MIDI clip in. **Undo** and **Redo** are on the **File** menu as well, grayed out when there's nothing to undo. Dragging a knob counts as a *single* step — one undo takes the whole move back, not one tiny nudge at a time. The history is per session: it starts fresh when you load or clear a song, and it isn't saved with the project.

Bars & chords

Add bars to the selected section with **+ Bar**. To put a chord in a bar, click its chord cell — that opens the **chord selector** (Chapter 9), which is how every chord gets entered. LeadMuse understands a wide vocabulary once you're in the selector or reading a chord back:

- Triads and sevenths: `C`, `Am`, `G7`, `Cmaj7`, `Dm7`, `F#m7b5`
- Extensions: `9`, `11`, `13` (e.g. `C13`, `Am11`)
- Slash chords / inversions: `C/E`, `G/B` — the part honors the slash note

More than one chord per bar

Set **Grid** to `1/8` or `1/16` and you can place chords on sub-beats instead of just the downbeat. Each chord holds until the next one comes along; an empty cell simply holds whatever chord came before it. If you're reading or hand-editing a song's plain-text file, a bar with more than one chord is written as `F@0 G@2` (F on beat 1, G on beat 3) or `Am@0 Dm7@1.5` (a change on the "and of 2").

Style overrides — a different player for a bar or a chord

By default, every bar plays in its section's style. Sometimes you want one passage to take a different voice while the rest of the section keeps its groove — a bridge that hands off to a busier player for four bars, say. You can override the style for a single bar, or even a single chord. An overridden bar or chord takes on *that*

style's own feel, so it genuinely sounds like the player you picked — and you can fine-tune that bar from there (see **Per-bar feel & variation** below).

- **Per bar** — click a bar number, in the row directly above the chord cells. This opens one combined menu for that bar, and style is the first thing on it: pick a style, or **Inherit section** to clear the override. That same menu is also where the per-bar **Role** override and the **Lick** markers live (Chapters 10 and 11) and where you mute the bar (below) — it's genuinely one menu, not several separate right-click actions. Overridden bars are tinted in the style's color.
- **Per chord** — right-click a chord cell and pick a style, or set the **Style** dropdown in the chord selector (Chapter 9). A chord override wins over both the bar and the section, and the cell is framed in the style's color. Choose **Inherit bar / section** to clear it.

Overrides are saved with the song and travel in shared `.lmsong` files. If whoever you send it to doesn't have that exact style installed, the bar just falls back to the section's style rather than breaking anything.

Per-bar feel — fine-tune one bar

A style override changes *who* plays a bar; you can also fine-tune *how* that bar plays, one bar at a time. Click a bar number and choose **Edit this bar's feel** from the same combined menu. The feel knobs above the grid switch to that single bar: turn any of them and it's pinned for that bar alone. The knob's label lights up to show it's overridden, and the bar gets a small accent underline along the bottom of its lane, so your tweaks read at a glance across the whole arrangement. A knob you haven't touched shows the value the bar *inherits*, so you always see what it will actually play.

What it inherits depends on the bar. If the bar carries a style override, its feel starts from *that style's* own settings — the same reason an override bar sounds like the player you gave it — so you're tuning the override, not the section. A plain bar inherits the section's feel. Either way, a knob you turn wins for that bar only.

Editing a bar's feel is an idle-time activity: press **Play** and the controls return to the live view, following the arrangement as usual. To stop editing a bar without playing, click any section band, or choose **Edit this bar's feel** again to toggle it off. **Reset this bar's feel** on the same menu clears a bar's overrides and returns it to what it inherits.

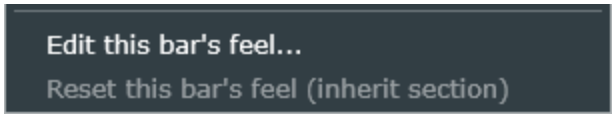

Figure 8.3 Editing one bar's feel — the feel knobs retarget to the focused bar, an overridden knob is lit, and the bar carries the accent underline that marks a per-bar override.

Silence — drop out a section or a bar

Sometimes the part should *lay out* — a rest bar, or a whole section that drops before a big return. You can silence either without deleting anything:

1. **Mute a section** — right-click the section band and choose **Mute Section (silence)**. The whole section goes silent; its band grays out and reads **MUTED**.

2. **Mute a bar** — click that bar's number and choose **Mute bar (silence)** from the same combined menu described above. The bar goes silent and its lane dims with an **M** badge.

Muting only ever affects *output*. The section or bar keeps its exact place in time — loop length and playhead are unaffected — and its chords, style, and feel are all kept intact, so un-muting brings back exactly what was there. A chord that was already holding when it runs into a muted bar simply resumes on the other side of it, and a note ringing into the drop gets trimmed cleanly rather than cutting off ugly. Mute is saved with the song, and both live playback and **Drag MIDI to your DAW** respect it. (Silencing an individual beat, rather than a whole bar, is planned for a later release.)

> **Tip.** Want the drop to still end with a pickup back in? Turn on **Keep fill when muted** in the same menu. The muted region still plays its transition fill — the run that leads into the next chord or section — while everything before it stays silent: the classic "drop, then a fill back in." The region shows as **MUTED (fill)** so you can see at a glance that the fill survives. It's off by default, so a plain mute is complete silence, and it only does anything where a fill is actually defined — a section's last bar, or a bar you've explicitly given a Fill marker.

Key

Leave **Key** on **Auto** and LeadMuse works out the key from your chords, shown as something like "Key: C major." Override it if you want to pin a specific tonal center — approach notes and walking lines respect whichever key is active, auto-detected or not.

Seed, octave & range

- **Seed** rolls a new variation while holding everything else fixed. Click the **dice** button beside it to jump to a random seed, the ▾ button to pick from the last 10 seeds you've used, or the **«** / **»** buttons to step through that recent list one seed at a time — LeadMuse remembers them, and they're saved with the song.
- **Octave** transposes the whole part up or down in octaves.
- **Range** (Low / High notes) folds every generated note by octaves to stay inside your instrument's playable window — useful when a sampled instrument has a limited range.

Range stays global to the whole song, but **Seed** and **Octave** don't have to be. Just like a section's time signature, you can give a single section its own — so a chorus can jump an octave, or take a completely different seed, while the verses stay exactly where they were. Change one of those controls while your song has more than one section and LeadMuse asks where it should apply:

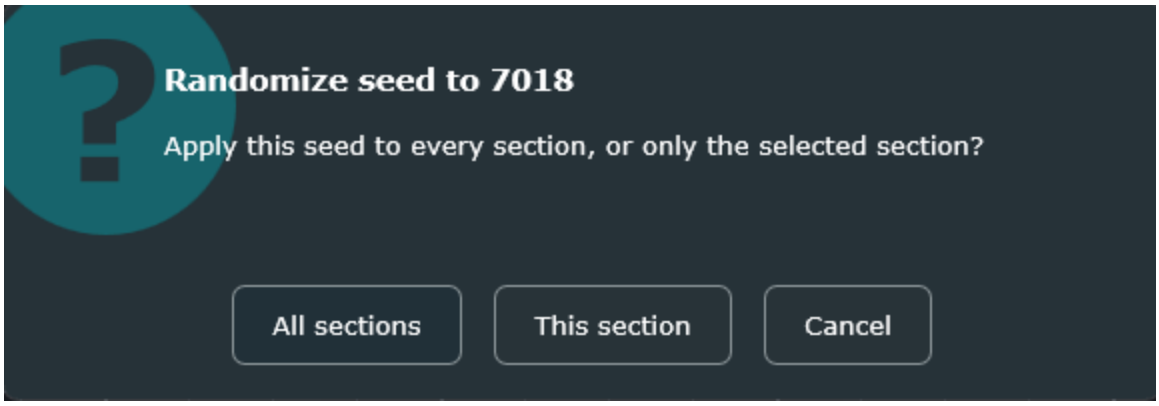

Figure 8.4 With more than one section, changing the seed asks where it lands. **All sections** sets the song-wide seed and clears any per-section seeds, so the whole song is back in step; **This section** gives just the selected section its own. (With a single section there's nothing to choose, so it simply applies.)

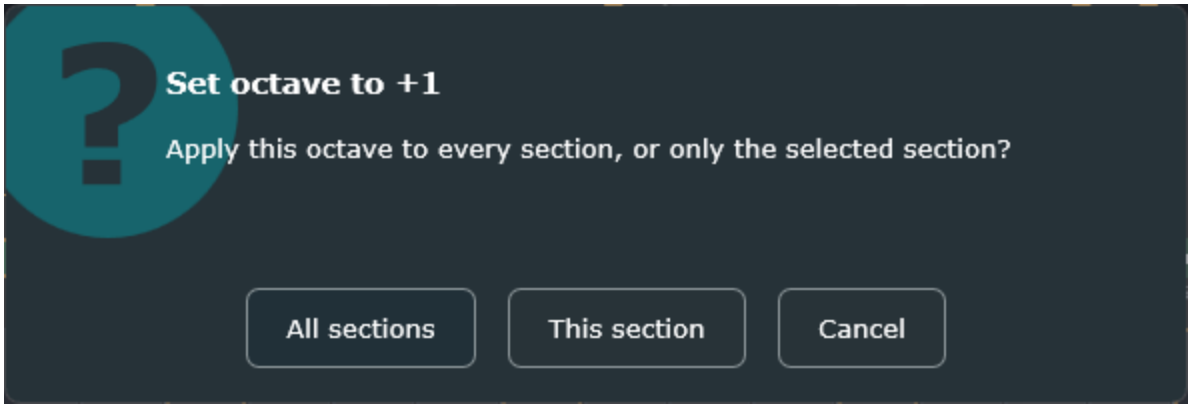

Figure 8.5 Octave works exactly the same way.

You can also set or clear a section's values from its right-click menu — each has an **Inherit** choice that hands it back to the song-wide value. A section that carries its own values shows them right on its band, so a glance tells you which sections differ: `#` with the seed, `O` with the octave shift.

Figure 8.6 A section with its own values shows them on its band — for example `#1234` and `O+1`.

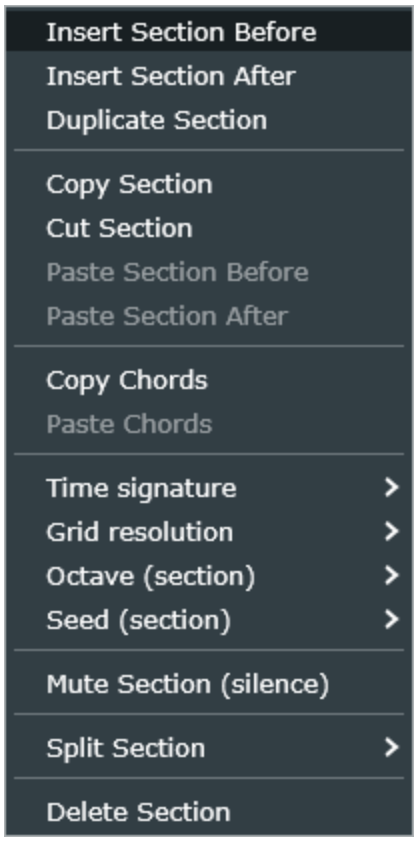

Figure 8.7 The section band's right-click menu sets the same per-section values — Time signature, Grid resolution, Octave and Seed each offer **Inherit** plus the per-section choices.

Live variation trigger

The **Seed** control re-rolls the *whole* song between takes — great for finding a part you like, but not something you'd want to touch mid-performance. The **variation trigger** does the same kind of re-roll live, per section, while the song is actually playing: a MIDI note you choose re-rolls the pattern for whichever section the playhead is currently in, and everything else in the arrangement is left untouched. It's built for performing — think of it as a "variation pad" you can tap between phrases.

Set it up under **File** → **Variation trigger**:

1. Click **Learn trigger note (tap your pad)**, then play the note you want to use. LeadMuse captures both the note *and* its MIDI channel, so whatever your controller or DAW happens to send just works — you don't need to guess the channel yourself.
2. Leave **Channel** on **Any**, or pin the trigger to one specific channel (1–16) if you'd rather keep it separate from other incoming MIDI.
3. Switch on **Enabled**. It's off by default, so a stray note on your controller can never re-roll a section by accident.

During playback, each tap steps to the next variation, and the new pattern always lands cleanly on the next downbeat — never mid-bar. It's deterministic: the same sequence of taps reproduces the same part every time, and your underlying song seed is never touched by any of this.

You'll see it happen, too. The section's band flashes the moment a variation fires, and stays marked with a small **▲N** badge for as long as it's sitting on a variation — **N** counts how many times you've re-rolled it, so a glance at the arrangement tells you which sections are on their base seed and which are running a variation. Found one you love? **File → Variation trigger → Bake current variations into song** folds it permanently into the saved song — it becomes that section's actual seed, survives edits and reloads, and clears the badge since there's no longer a "variation" to distinguish it from. This all works in **Song (grid)** mode, since that's the mode where LeadMuse owns the sections in the first place.

> **Tip.** Hover any section's band to read out its **effective seed** and exactly how it's built — base seed, plus any baked bump, plus the current variation offset — so when a re-roll sounds great, you can write the numbers down and reproduce it later. And if you'd rather keep your hands on the controller than the mouse, **File → Variation trigger → Learn step-back note** assigns an optional MIDI note that steps a variation backward — the companion to the trigger note, which only ever steps forward.

9 The chord selector

The chord selector is a small window that lets you build a chord by clicking instead of spelling it out. It's the same idea as a chord chart taped to a music stand — you look at the shape you want, you point at it, and you move on. The difference is that this chart also shows you the song key, lights up the notes on a keyboard, and writes the finished symbol into your arrangement for you.

This matters because LeadMuse builds its bass line from the chord symbols in the grid, and a symbol it can't read is a symbol it quietly ignores. The selector never hands you an unreadable chord. Every chord you build here is one LeadMuse can play.

You'll spend more time in this window than almost any other, so it's worth ten minutes learning what each part does. Let's open it first.

7.1 Opening the selector

The selector belongs to the arrangement grid, so you reach it by clicking the cell you want to change.

1. Set **Source** to **Song (grid)**. The arrangement grid appears below the transport controls.
2. Find the bar you want to work on and click its chord cell — the row of boxes underneath the bar numbers.
3. Notice how the rest of the window dims and the selector appears in the middle. That dimmed layer is deliberate; it tells you the selector has your attention until you're done with it.

Your screen should now look like the one shown in Figure 9.1.

Figure 9.1 The chord selector. The wheel's center reads "C major", the Am segment on the inner ring is filled, and the preview at the bottom left confirms the chord as `Am`.

If the cell already had a chord in it, the selector opens with that chord loaded: the root, the type and the slash bass are all set for you. An empty cell shows a small dash, and clicking it opens the selector on **C major** — the starting point the window uses when it has nothing to go on.

> **Note.** The selector is drawn *inside* the plugin window rather than as a separate floating window. That's why it behaves properly in Logic Pro and other hosts that keep plug-in windows penned in. If your plugin window is too small to hold it, LeadMuse grows the window temporarily and puts it back to its original size when you close the selector.

> **Note.** Right-clicking a chord cell does something different — it opens the per-chord style menu, not the selector. If nothing happened when you right-clicked, the cell was empty; there's no chord there to give a style to.

The selector is a Song (grid) feature. In **Follow MIDI** mode there's no arrangement to edit, so the grid — and with it the selector — isn't on screen at all.

7.2 What you're looking at

The window is laid out in four regions, and once you can name them the rest of this chapter is easy going:

- **The circle of fifths**, in the middle. This is where you pick the root note.
- **TYPE**, down the left side. Seven buttons that set the basic flavor of the chord.
- **ROOT**, **INTERVALS**, **BASS** and **STYLE**, stacked down the right side. Fine control over the root, the color tones, the slash bass and the bassist style.
- **The preview and the piano**, along the bottom. The chord symbol you've built so far, the **Apply** and **Clear** buttons, and a two-octave keyboard showing the notes.

Everything updates live. Click anything at all and the preview text, the highlighted keys and the highlighted wheel segment redraw together, so you're never guessing what you're about to write into the song.

7.3 Picking a root from the circle of fifths

The circle of fifths is a way of arranging the twelve notes so that neighbors sound related. Going clockwise from C you get G, D, A and so on — each one a fifth up from the last, and each one a close musical relative of the one before it. Songwriters have leaned on this layout for a couple of centuries, because chords that sit near each other on the circle tend to sound good near each other in a progression.

LeadMuse draws it as two rings with the song key in the middle:

- **The outer ring** holds the twelve major roots.
- **The inner ring** holds the relative minors — the minor chord that shares a key signature with the major directly outside it.
- **The center** shows the song's key, such as "C major". It reads `-` when LeadMuse hasn't got enough chords in the song to work a key out yet.

To use it, click a segment:

1. Click a segment on the outer ring. Notice how the segment fills with the accent color, the **TYPE** buttons jump to **Major**, and the preview at the bottom shows the new symbol.

2. Click a segment on the inner ring instead. This time you get that segment's minor chord, and **TYPE** switches to **Minor**.
3. Look at the center before you pick. Roots near your key are the ones most likely to sound settled.

Figure 9.1, back in the previous section, shows step 2 in progress. The song is in C major, and the Am segment on the inner ring has been clicked: it fills with the accent color, **TYPE** switches to **Minor**, and the preview reads `Am`. Notice that Am sits in the same wedge as C on the outer ring — that's the relative minor, the closest possible neighbor to the song's own key.

The center isn't clickable, and neither is the area outside the wheel — clicks land only on the two rings. So don't be afraid to explore; the worst a stray click does is nothing at all.

> **Tip.** I use the wheel for the root and the type, then switch to the right-hand column for everything else. The wheel is faster for the decision that matters most, and the dropdowns are more precise for the trimmings.

If you'd rather not use the wheel, the **ROOT** dropdown beside it does the same job. It lists all twelve notes, and it always agrees with the wheel — pick from one and the other follows.

> **Note.** Root and bass note names are written with sharps: C#, D#, F#, G#, A#. There's no flat spelling in these lists, so where you'd write E-flat you'll pick D# instead. It's the same note, and LeadMuse plays it the same way.

The center readout comes from the **Key** control on the main window. Leave that on **Key: Auto** and LeadMuse works the key out from every chord in the song; set it explicitly and the wheel's center shows your choice instead.

7.4 Setting the chord type

The **TYPE** buttons set the basic quality of the chord — the part that decides whether it sounds bright, dark or unsettled. There are seven, and the selected one is highlighted in the accent color.

Button	What it writes	What it sounds like
Major	`C`	The plain, bright triad
Minor	`Cm`	The darker triad
Dim	`Cdim`	Tense and unresolved — a passing chord
Aug	`Caug`	Raised fifth; restless
sus2	`Csus2`	Open and unresolved, with the third replaced
sus4	`Csus4`	The classic suspension that wants to fall
Power	`C5`	Root and fifth only; neither major nor minor

Only one type can be active at a time, which is exactly right — a chord is either major or minor, not both. Click a different button and the previous one releases.

Two of these deserve a word of honesty before you build a song around them.

> **Warning.** LeadMuse's underlying chord-quality model — shared by both products — doesn't carry an augmented or a sus2 quality of its own. **Aug** is generated as a major chord, and **sus2** is generated as a sus4. The chord MIDI drag-out does voice the raised fifth for **Aug**, so exported chords are correct — but if the raised fifth or the second is the whole point of the passage, listen to the generated part before you commit.

7.5 Adding color with the interval buttons

The **INTERVALS** buttons add the notes that turn a plain triad into something with character. A C major triad is three notes; add the seventh and you get `Cmaj7`, which is the same chord wearing a nicer jacket. These are the tones players call *extensions* (the 6, 7, 9, 11 and 13) and *alterations* (the flattened and sharpened ones).

There are twelve buttons in a three-column grid:

- **Extensions:** 6, 7, maj7, 9, 11, 13
- **Alterations:** b5, #5, b9, #9, #11, b13

They're toggles rather than a single choice. Click one to switch it on, click it again to switch it off, and the ones that are on stay highlighted. Switch on as many as you like.

Here's the part that surprises people, so I want to be straight about it. The chord symbol keeps only **one** extension, even when several buttons are lit. LeadMuse picks it in this order and uses the first one it finds: maj7, then 13, then 11, then 9, then 7, then 6. So if you turn on both 7 and 9, the preview reads `C9`, not `C79` — which is right, because in real chord notation a 9th chord already contains the 7th.

The alterations follow their own rule, and it's worth knowing why. An alteration only reaches the symbol when the chord already has a seventh or higher in it — that is, when maj7, 13, 11, 9 or 7 is switched on. Click **b9** on a plain major triad and the preview won't change, because "C with a flat ninth and no seventh" isn't a chord anyone writes.

1. Click **Major** in the TYPE column, then click **b9**. Notice that the preview still reads just `C`.
2. Now click **7**. The preview jumps to `C7b9` — the alteration finally had somewhere to attach itself.
3. Click **7** again to switch it off. The preview drops back to `C`, taking the alteration with it.

> **Note.** If you build a combination that isn't a real chord — a diminished chord with a major seventh, say — LeadMuse falls back to the plain triad rather than writing something it can't play. The preview shows you the fallback immediately, so you'll see it happen.

> **Note.** One chord you can't build here is a minor-major seventh (`CmMaj7`). Selecting **Minor** and **maj7** together gives you plain `Cm`, because the maj7 marker is ignored when the type is minor.

7.6 Choosing a slash bass

The **BASS** dropdown puts a different note underneath the chord — what's written as `C/E` and read aloud as "C over E". That bottom note is what your ear hears as the foundation, so changing it changes the whole feel

of the bar without changing the harmony one bit. It's the standard way to write an inversion, and it's how you get a bass line that walks down while the chords stay put.

The list starts with an em dash (—), which means "no slash bass, just play the root," followed by the twelve notes. Pick a note and the preview appends it after a slash; pick the dash to take it away again.

LeadMuse honours the slash note in the generated part, which is the whole point. A `C/E` tells the bassist to sit on E, not C.

> **Tip.** Try `C`, `G/B`, `Am`, `Am/G` across four bars. The bass walks C–B–A–G under chords that barely move, and it's one of the most-used moves in popular music.

7.7 Reading the piano preview

The keyboard along the bottom shows the chord you've built, highlighted across two octaves. LeadMuse voices the chord once — it doesn't light up every octave — and shifts the whole voicing to sit near the middle of the keyboard, so the shape you see is roughly the shape a player's hand would make.

Use it as a sanity check rather than a score. Glance at it before you click **Apply** and you'll catch the wrong-root and wrong-quality mistakes that are otherwise invisible in a chord symbol.

The preview does have limits, and knowing them saves confusion later:

- It shows the chord tones plus any of **6**, **9**, **11** or **13** you've switched on, even when the symbol kept only one of them.
- It does **not** show the alterations. A `C7b9` lights up as a plain C7 with the perfect fifth in place, so the flattened ninth won't appear even though the symbol carries it.
- The slash-bass note is included in the highlighted set.

> **Note.** The keyboard is a display only. Clicking it doesn't do anything and doesn't play a sound.

7.8 Giving one chord its own style

The **STYLE** dropdown is the odd one out in this window: it doesn't change the chord at all. It changes *who plays it*. Every chord normally inherits its style from its bar, and the bar from its section — but sometimes you want one chord in a turnaround to come from a different player entirely.

Leave it on **Inherit bar / section** and everything behaves normally. Pick a style from the list and that one chord plays in that style, overruling both the bar and the section. The section's feel knobs — density, swing, ghost, legato and the rest — still apply; only the player changes.

You'll see the override on the grid straight away. The cell picks up a colored wash and a bold frame in the style's color, so one scan of the arrangement tells you which chords have been singled out.

> **More Info.** Per-bar and per-section style overrides, and the order they win in, are covered in Chapter 8 under "Style overrides."

7.9 Applying, clearing and backing out

You've got three ways to leave the selector, and they do genuinely different things.

1. Click **Apply** to write the chord into the cell. The chord symbol and any style override you picked both go in, the selector closes, and the cell in the grid shows your chord.
2. Click **Clear** to empty the cell. The chord is removed and so is its style override, the selector closes, and the cell shows a dash again.
3. Click anywhere on the dimmed area outside the selector to back out. Nothing is written, and the cell keeps whatever it had before.

Get into the habit of using the dimmed backdrop when you're just browsing. It's the only exit that changes nothing, and it saves you from talking yourself into a chord you were only auditioning.

> **Warning.** An emptied cell isn't a silent bar. When a cell has no chord, the previous chord simply holds through it — that's how you write a chord that lasts four bars. If you want silence, mute the bar instead (Chapter 8).

7.10 Placing more than one chord in a bar

One cell holds one chord, so the number of cells per bar decides how finely you can place changes. That's the job of the **Grid** dropdown beside the arrangement controls: **1/4** gives you one cell per beat, **1/8** gives you two, and **1/16** gives you four.

To put a chord on the "and" of beat 2, set **Grid** to **1/8** first, then click the cell you want. Each chord holds until the next one, so you only need to place a chord where something actually changes.

> **More Info.** Grid resolution, including the per-section override, is covered in Chapter 8. For now, think of it as a view setting — changing it never changes playback, only how precisely you can place a chord.

7.11 One thing to watch when you reopen a chord

Reopening a chord doesn't always bring back everything you put into it. LeadMuse stores the chord as a symbol, and when the selector reads that symbol back it restores the root, the base quality and the slash bass — but not the extensions and alterations.

Here's what that means in practice. Say you build `C13` and apply it. Click that cell again and the selector opens showing **Major** with the **7** button lit, and the preview reads `C7`. The 13 is still in your song, but it isn't in the window any more. Click **Apply** at that point and you'll overwrite `C13` with `C7` without meaning to.

> **Warning.** If you open a chord that has extensions or alterations and you didn't intend to change it, back out by clicking the dimmed area rather than clicking **Apply**. Applying rewrites the cell with whatever the window is showing, which may be a simpler chord than the one you started with.

The same thing happens with **Aug** and **sus2**, which come back as **Major** and **sus4** respectively. There's no way around this from inside the window, so the habit to build is a simple one: if you're only looking, click the backdrop.

I don't want to make this sound worse than it is. It only bites on the richer chords, it's completely predictable once you know about it, and the preview always tells you exactly what **Apply** is about to write. Read the preview line, and you'll never be caught out.

10 Styles & the feel knobs

A **style** is LeadMuse's model of a player's characteristics — their register, their rhythmic feel, their harmonic vocabulary, and how they articulate a note. It's not a transcription of anyone real; think of it more like a character an actor plays than an impersonation of a specific person. That's deliberate: it means every style is legally LeadMuse's own, and it also means you're picking a *sound*, not a copy of a recording.

Ten lead-guitar styles are built in. Each one models a different lead player's approach — the register they live in, their phrasing, and the mix of bends, vibrato and fast runs (Chapter 11 covers the lead engine behind all of it):

Style (dropdown label)	Character
Blues Lead (B.B. Kingpin)	Singing and sparse, with huge bends and slow vibrato
Slide Rider (Dew Allbran)	Gliding legato slide lines, bluesy, heavy vibrato
Acid Rock (Jimi Hendhaze)	Expressive psychedelic rock — bends and a chromatic edge
Hard Rock (Jammy's Pages)	Pentatonic hard-rock riff-solos with attitude
Shred (Eddie Van Wailin')	Fast scale runs and big leaps, up in the high register
Latin Rock (Carlos Santosa)	Singing, sustained melodic lines with rich vibrato
Country Lead (Chet Atkinson)	Bright hybrid-picked lines, wide intervals and twang
Funk Lead (The Artist)	Punchy, syncopated, short rhythmic phrases
Rock'n'Roll (Chuck Bury)	Driving major-pentatonic riffs and repeating motifs
Cello (MoMoYa)	Ultra-legato, lyrical sustained lines — rich vibrato, no string bends, lower register

The feel knobs

Picking a style loads a full set of defaults into the feel knobs, so you always start from a sensible place — a Jazz style starts swung and walking, a Synth-Pop style starts stiff and metronomic. From there, the knobs are yours to bend. They're per-section, so a Verse and a Chorus can breathe completely differently even when they share a style.

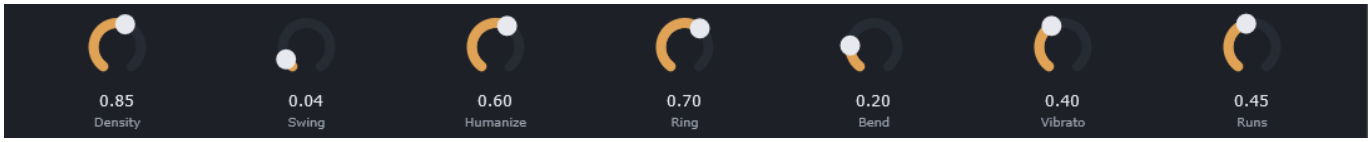

Figure 10.1 The feel knobs, showing the values the selected style loaded. Every knob is per-section; the table below is what each one does.

Knob	Range	Effect
Density	0–1.5	How busy the part is — more notes vs. more space
Swing	0–0.5	Delays the off-beat for a shuffle feel

Humanize	0–1	Loosens timing and velocity so it feels played, not typed
Ring	0–1.1	Note length — from short and detached to long, held and connected into the next note
Bend	0–1	How often a held note bends up a whole step and settles back — a real pitch-bend your target instrument plays. Each style sets its own default.
Vibrato	0–1	How often a held note gets a vibrato swell, sent as CC1 (the mod wheel). Each style sets its own default.
Runs	0–1	Fast-run tendency — low is sparse, spacious phrasing; high reaches for longer, busier strings of notes. Each style sets its own default.

That's seven knobs on LeadMuse. There's no Complexity, Ghost, Flow or Fill knob here — a lead line is a single voice, so those make no sense; the **Role** control and the **Lick** markers (both in Chapter 11) do the job those would have done on the other plugins.

Per-bar and per-beat Lick markers

LeadMuse has no Fill knob. Instead of a run that only fires at the section end, you drop a **lick** — a short placed phrase — exactly where you want one. Click that bar's number, in the row above the chord cells, and the menu that opens offers two Lick entries:

- **Lick (this bar)** places a lick at the bar's downbeat, at **None**, **Light**, **Medium** or **Heavy** intensity — intensity sets how long and busy the phrase is.
- **Lick on beat** drops one on a specific beat instead. A lick on the last beat is the classic run into the change.

A lick plays whatever the section's Role is — even **Off** — and owns the line from its beat to the end of that bar. Marked bars show a small amber **L** badge, with a dot on each beat that carries a lick; the markers save with the song and travel in your `.lmsong` file and DAW session. Role, licks and the lead expression all get the full treatment in Chapter 11.

> **Note.** Those Lick entries aren't a separate right-click destination — they're submenus inside the same combined menu you get from clicking a bar number at all, alongside the per-bar **Role** override, the per-bar style override and the mute toggle covered in Chapter 8. One click on the bar number gets you all of it.

> **Tip.** Hover any knob, button, or menu for a one-line reminder of what it does. If you'd rather see a cleaner window, turn the hints off with **File → Show tooltips**.

> **Note.** LeadMuse checks for a newer release each time it loads, and shows an amber **Update to v...** button in the header when one's available. If you'd like to help test prereleases, tick **File → Get beta updates** — the button then reads **Update to v... (beta)** and follows the beta channel instead of stable releases.

You can also build your own styles as plain-text `.lmstyle` files — Chapter 13 and the Appendix cover the format.

11 Role, licks & expression

LeadMuse is a lead engine. Where a rhythm-guitar part is a hand working a pick across a voiced chord, a lead is the opposite shape entirely: *one note at a time*, a single voice phrasing a melody over the chords rather than stating them. Everything in this chapter serves that single line — how much of it plays (**Role**), where you drop a placed phrase (**Licks**), and the bends and vibrato that make a held note sing (**expression**). Picking a style sets the register, the phrasing, and how much of that expression comes naturally; these controls shape it from there.

Role — how much it plays

The one control unique to LeadMuse sits right beside the style menu: **Role**. It's a single ladder from silence to a full solo, and it's the first thing to reach for when a section is playing too much or too little.

Role	What the line does
Off	Silent — the section lays out. (Any Licks you've placed still fire; see below.)
Lead: Low	One voice-led note per chord, on its beat — a spare, chord-anchored melody
Lead: Med	Adds on-beat movement within each chord — a little more to say, still anchored
Lead: High	A note on every beat — a flowing, quarter-note melodic line
Solo	The free phrasing engine — fast runs, leaps and licks separated by space, up in the register

Think of it as how forward the player is leaning. The three **Lead** rungs stay close to the chords, filling in more movement as you climb; **Solo** cuts loose. Role is per-section, and like style it can be overridden on a single bar — a quiet verse can hand one bar to the soloist and take it straight back, from the same combined bar-number menu as everything else (Chapter 8).

Licks — placed phrases

A **lick** is a short phrase you drop at an exact spot, independent of the section's Role — the lead analog of the Fill marker on the other Muse plugins, but finer. Chapter 10 covers placing them from the bar-number menu, as a whole-bar lick or on a specific beat, at Light / Medium / Heavy intensity. Two things are worth repeating here, because they're what makes the feature musical:

- **Licks fire at any Role, even Off.** So you can rest a section entirely and still punch in a single answering phrase exactly where it belongs — the sparsest, most tasteful way to use a lead.
- **A lick owns the line from its beat to the end of its bar.** Whatever the Role was doing there steps aside, the lick plays, and the line picks back up in the next bar. A lick on the last beat becomes the classic run into the change.

Expression — bends, vibrato and real MIDI

A lead lives or dies on expression, so LeadMuse writes it as proper MIDI rather than faking it with velocity. A held note can **bend** — a whole-step pitch-bend up into the note, released back to pitch so it never hangs — and a longer note can take a **vibrato** swell, sent as CC1 (the mod wheel). How often each happens is part of

the style, and the **Bend** and **Vibrato** knobs (Chapter 10) scale it to taste; the **Runs** knob pushes the line toward longer, busier phrases.

> **Important.** The bends are real channel pitch-bend, so your target instrument has to agree on how far a full bend reaches. LeadMuse bends a **whole step (2 semitones)**, which is the standard default — but if a guitar instrument is set to a different pitch-bend range, the bends will come out too wide or too narrow. If a bend sounds wrong, set the instrument's pitch-bend range to **±2 semitones** and it will track exactly.

> **Note.** The built-in **Preview** (Chapter 14) auditions through a sampled guitar that doesn't voice pitch-bend, so Preview won't reproduce the bends — they're there in the MIDI and you'll hear them through your routed instrument and in the dragged-out clip. Preview is for checking notes and timing; your own VSTi is where the expression lands.

How the line chooses its notes

Under all of it is one rule: the line is **monophonic** and connected. On a chord's beat it lands on a chord tone, voice-led from wherever the previous note left off so it moves by the smallest sensible step; between those anchors it steps through the scale, resolving back onto chord tones so the phrase stays inside the harmony even as it runs. The line flows *across* bar and chord changes rather than restarting each bar, and because only one note sounds at a time, each note is trimmed so it never overlaps the next — which is also what lets a held note ring right up to the following attack.

How the feel knobs map to the lead

The feel knobs (Chapter 10) drive this engine directly. **Density** sets how much space sits between phrases when the line is soloing. **Swing** pushes the off-beats. **Humanize** loosens timing and velocity so it plays like a hand. **Ring** (the "Legato" knob elsewhere) sets note length, from short and detached to long and connected. And **Bend**, **Vibrato** and **Runs** are the three lead-only knobs described above.

> **Tip.** As with the other plugins, the instrument you feed matters — but for LeadMuse it's specifically how it handles *expression* that counts. A patch that responds smoothly to pitch-bend and mod-wheel vibrato will sing; a stiff one will merely play the notes. Set its bend range to ±2 semitones, map CC1 to a little vibrato depth, and the lead comes alive.

12 Drag-MIDI — export & import

Every DAW can accept a MIDI clip dragged onto a track. That's the one thing you can always count on, no matter how unusual your routing is or how a particular host feels about hosting LeadMuse as a MIDI effect — which is exactly why LeadMuse builds its export and import around plain drag-and-drop instead of anything more elaborate.

Dragging your part out

The drag strip at the bottom of the window prints your whole arrangement to a standard `.mid` file the moment you start dragging it. Press and drag it straight onto a track in your DAW — or into a folder, if you just want the file — and let go. You get back a MIDI clip of the part, ready to use.

Figure 12.1 The two drag strips at the bottom of the window — . Press and drag either onto a DAW track.

This is the **universal fallback**. It works in every DAW, with no routing to set up and no plugin-format concerns, and it bounces down perfectly because by the time it hits your track, it's just MIDI — nothing depends on LeadMuse being loaded at mixdown time. Reach for it whenever live routing feels like more setup than the moment calls for, or any time you're working in Pro Tools (Chapter 15), where it's one of only two ways in.

> **Note.** Because the engine is deterministic (Chapter 1), the MIDI you just dragged out is *exactly* what you heard — same seed, same notes, not an approximation of it.

Drag Chords — a chord track for piano or pad

Right beside the part strip is a second one that prints the arranger's chords themselves, as sustained block-chord MIDI, instead of the generated part. Drag it onto a piano, pad, or synth track to comp underneath whatever LeadMuse is playing. It voices the full chord — extensions and all — as an ascending spread starting around C3, so a dense 11th or 13th chord opens outward instead of clumping into a cluster down low.

By default, that voicing leaves the very bottom out — **File → Chord drag: omit low root** is on out of the box, on the theory that the generated part already owns the low end and doesn't need company down there. Turn it off and a low root gets added below the voicing to anchor it on its own, useful if you're dragging chords out without a bass part alongside them. Your choice here is saved with the song.

Just the selected section

Both drag strips normally export the whole song, but the **Selected section only** toggle next to them scopes both at once to whichever section you currently have selected — rebased to start at bar 1, so the clip you get is just that section, standing on its own. The strip labels update to show which section you're about to drop, so you're never guessing.

Generate chords from dropped MIDI

The strips work in reverse too: drag a MIDI clip *out of* your DAW and drop it onto the arranger, and LeadMuse reads the harmony out of it and writes a chord progression into the grid for you. A few things make this more reliable than it might sound:

- If your clip has more than one track, LeadMuse automatically reads from whichever one is the most "chordal" — the one with the most notes actually stacked together — so a bass or melody line sitting alongside your chords doesn't throw off the detection.
- It samples the harmony at whatever your **Grid** resolution is currently set to, and collapses consecutive repeats, so you don't end up with the same chord written into every single cell.
- It matches the section's meter to the clip. If the clip carries a time signature — most DAW exports do — LeadMuse sets the target section to that meter first, so a 7/8 (or 5/8, 9/8) progression lands one chord per bar instead of drifting against a 4/4 section.
- Some tools export chords without a time signature — **Scaler** is the one you're most likely to meet. With no meter to read, LeadMuse looks at how the chords are *spaced* instead: evenly spaced odd-eighth bars (3½, 2½, or 4½ beats) are read as 7/8, 5/8, or 9/8; chords sitting on whole beats stay a plain 4/4, since that's what the clip actually contains. One thing to know: depending on the DAW Scaler is hosted in, a 7/8 part may come out at its true 3½-beat spacing — which LeadMuse catches — or spaced as plain 4/4, in which case the odd meter simply isn't in the file for anything to recover. If an odd-meter drop lands as 4/4, set the section's **Time signature** by hand and drop the clip again.

When you drop the clip, a menu asks where the chords should land:

Figure 12.2 The drop-target menu. Each option names the section it would affect, so you can see exactly where the chords will go before you commit.

1. **Whole song** — split across every section by position; the last section grows if the clip runs longer than the song.
2. **Selected section (name)** — replace whichever section you currently have selected.
3. **This section (name)** — replace the section you actually dropped the clip on.
4. **From this measure (bar N)** — fill forward from the bar you dropped on toward the end of that section, leaving everything before it untouched.
5. **As a new section (at the end)** — create a brand-new section for the chords. LeadMuse asks for its time signature first, so a 4/4 clip dropped after a 3/4 section still gets a proper 4/4 home — and the new section inherits the previous one's style and feel, rather than starting from a default.

If you picked a section or a from-this-measure target and the clip is longer than the room it has, LeadMuse asks — by name, with the real numbers on both sides — something like "*The dropped chords span 12 bars, but this section holds 8. Clip the extra bars, or extend this section to fit?*", with **Clip to fit** and **Extend this section** as your two options. Whole Song and As A New Section never need to ask — they simply grow to fit. And no matter which target you pick, if it would overwrite chords that are already there, LeadMuse asks you to confirm the replacement first, naming exactly what's about to be replaced ("*this section,*" "*the whole song,*" or "*this measure onward*").

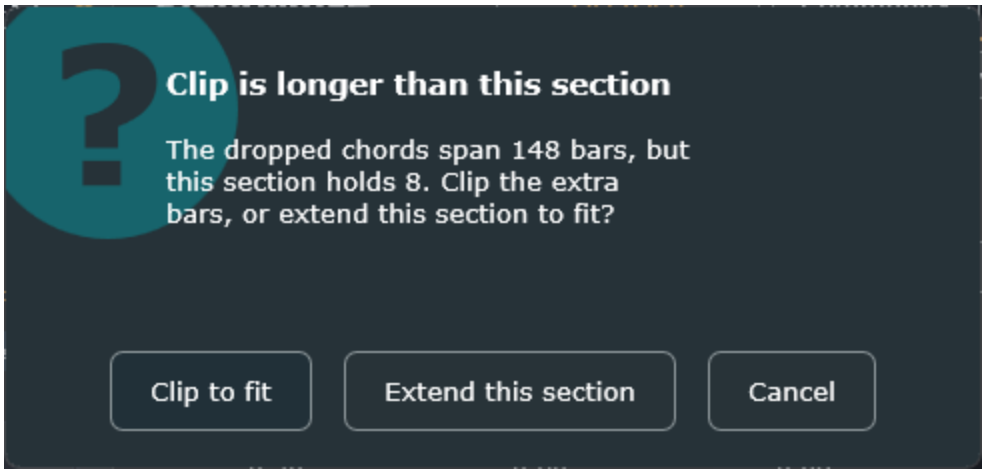

Figure 12.3 The fit prompt — shown only when the clip has more bars than a bounded target. Clip the extra bars off, or grow the section to hold them.

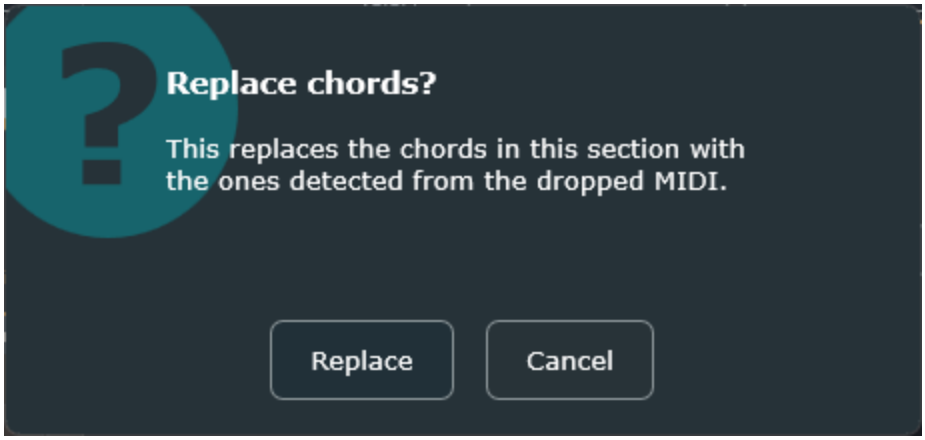

Figure 12.4 The replace confirmation — shown only when the target already holds chords, so a drop never silently overwrites work you'd done.

Capture chords live from MIDI

Not every DAW will let you drag a MIDI clip onto a plugin's window — so for those, LeadMuse can read your chords as you actually play them instead.

1. Turn on **File** → **Capture chords from MIDI**. LeadMuse arms itself, ready to listen.
2. Route a chord track — or your keyboard — into LeadMuse's MIDI input.
3. Play. Each chord you play is timestamped against the transport as it happens.

4. Stop the transport. LeadMuse automatically disarms and writes what you played straight into the grid — the same end result as a drop, without needing to drag anything at all.

LeadMuse still asks before replacing chords that are already there, exactly as it does for a dropped clip.

13 Sharing: styles, songs, bundles

Everything LeadMuse saves is plain text. There's no proprietary pack format, no samples riding along, and nothing to recompile — a style, a song, or a whole bundle is just a text file you can open, read, email, or drop into a chat. That's a deliberate choice: it means sharing your work is exactly as easy as sharing any other document.

There are three file types, and each one exists for a different-sized job:

File	Contains	Save / load with
`.lmstyle`	One style	Import Style / Export Style
`.lmsong`	A whole arrangement — sections, meters, chords, styles referenced by id	File ‣ Save / Load Song
`.lmbundle`	A song *plus* every style it uses — fully self-contained	File ‣ Save / Load Bundle

Cross-product songs

File ‣ Load Song doesn't just open LeadMuse's own song files — it also opens the song files from BassMuse, PianoMuse and RhythmMuse. The arrangement transfers completely: sections, chords, time signatures, per-bar fills, and mutes all come across, and LeadMuse applies its own styles to it, since every Muse plugin shares one underlying song format. Practically, that means a song built in one plugin is never locked out of another — an arrangement you wrote for one instrument can become the basis for a part on another, without retyping a single chord.

Copy from a running plugin

When you have more than one Muse plugin open in the same session — say a bass part on one track and a piano part on another — LeadMuse can pull an arrangement straight from a neighbour, with no file in between. Open **File ‣ Copy arrangement from a running plugin** and pick any other Muse instance that's currently loaded; its *current* arrangement is copied into LeadMuse on the spot.

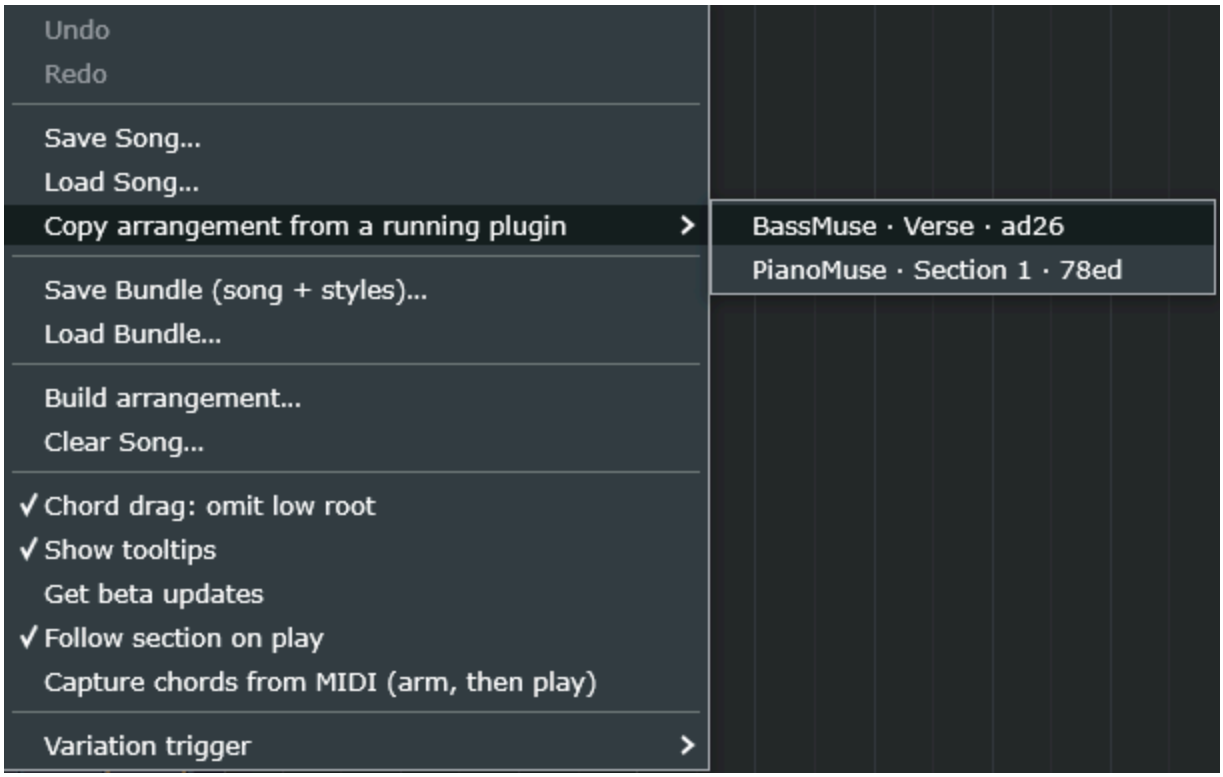

Figure 13.1 Each running Muse plugin lists the others by product, section name and a short id — pick one and its arrangement is copied in.

It's the same transfer as opening another product's song file: sections, chords, meters, per-bar fills and mutes all come across, LeadMuse applies its own styles, and the note range is reset to suit this instrument. The copy is a one-time snapshot, not a live link — edit it freely afterwards without touching the plugin you copied from. Each instance keeps its neighbours posted every few seconds, so the list always reflects what's open right now.

Where files live

LeadMuse treats styles and songs differently, and it's worth knowing why:

- **Styles** live in `Documents/LeadMuse/Styles/*.lmstyle`. LeadMuse scans this folder once, at startup, and adds anything it finds to the Style dropdown automatically — so a style someone shares with you only needs to land in that one folder to show up ready to use.
- **Songs** default to `Documents/LeadMuse/Songs/*.lmsong`, but that's only ever a starting suggestion in the file dialog — you can save a song anywhere you like. Songs aren't scanned or auto-loaded the way styles are; you always load one explicitly.

> **Note.** Portability. Songs reference their styles by a short internal **id**, not by copying the style's data — that's what keeps a `.lmsong` file small and easy to share. If you send someone a song and they're missing one of its custom styles, that section simply falls back to the first style in their list; the chords, meter, and feel are completely unaffected, so the song still plays, just possibly with a different voice on that section than you intended. Send a **bundle** instead whenever you want to guarantee they get everything, in one file, with nothing missing.

Tidying up the style list

Because LeadMuse adds every style file it finds to the dropdown, a stretch of happy experimenting can leave the list cluttered with versions you no longer want. There's no delete button inside the plugin — and that's on purpose, so a stray click can never wipe out work — but clearing out the old ones is quick, because each style is just a file you can throw away.

1. **Open your Styles folder.** On **Windows** it's `Documents\LeadMuse\Styles`; on **macOS** it's `~/Documents/LeadMuse/Styles` (in Finder, use **Go** ▸ **Home** ▸ **Documents** ▸ **LeadMuse** ▸ **Styles**, or press **Cmd+Shift+G** and paste the path).
2. **Delete the ones you don't want.** Each experiment is a single `.lmstyle` file named after the style, so you can keep the keepers and bin the rest — drag them to the Recycle Bin on Windows or the Trash on a Mac.
3. **Reload LeadMuse.** Remember that LeadMuse scans the folder only once, at startup, so the dropdown won't change until a fresh instance loads: remove and re-add the plugin on the track, or just restart your DAW. The names you deleted are now gone from the list.

> **Note.** The built-in styles live inside the plugin itself, not in this folder, so they're always present and this never touches them — only your own saved and imported styles come and go with the files here. And if there's an experiment you'd rather keep, the same folder is where it lives, so it's easy to copy somewhere safe before you start deleting.

Rolling your own style

A `.lmstyle` file is a small, hand-editable text format: `key = value` lines, with `#` starting a comment. A rhythm figure is written as `cell = <onset ...> | <weight>`, and a bar's accent pattern as `accent = a b c d`. Between those lines, you control a style's register, its swing, the weights behind its note choices, its velocity and articulation, and its overall feel — everything that makes one style sound different from another. The Appendix has the full key list if you want to build one from scratch, or just want to understand what a line in someone else's style file is actually doing.

14 Using LeadMuse in your DAW

LeadMuse's Follow-MIDI code never changes from one DAW to the next — it just reads note-on and note-off from whatever MIDI arrives at its input, the same way in Reaper as in Cubase. What *does* change, DAW by DAW, is **routing**: how each host actually lets a MIDI-generating plugin sit between your chords and an instrument. Once you know which of three patterns your DAW uses, the rest is just mechanics.

- **Model A — in-chain MIDI FX.** LeadMuse and the instrument live in the same track's device chain. This is the cleanest pattern by far: chords go in one end, a finished part comes out the other, and there's nothing to route by hand.
- **Model B — instrument + MIDI-out route.** LeadMuse is loaded as an *instrument* on one track, and its MIDI output is routed to a second track that hosts the real instrument. Some hosts simply won't let a third-party plugin sit inline as a MIDI effect, so this is the workaround.
- **Model C — format-incompatible.** The DAW can't load LeadMuse's plugin format at all. Today that's Pro Tools, which speaks AAX only (Chapter 15).

DAW	Format	Model	Routing in a nutshell
Reaper	VST3 / AU	A	LeadMuse, then the instrument, in one track's FX chain
Bitwig	VST3	A	LeadMuse before the instrument in the device chain
Logic Pro	AU	A	LeadMuse in the track's **MIDI FX** slot
Mixbus 11	VST3 / AU	A	LeadMuse then the instrument in the processor box
FL Studio 2024	VST3	A	Wire chords → LeadMuse → synth inside **Patcher**
Waveform 14 (Tracktion)	VST3 / AU	A	LeadMuse then the instrument in one track's plugin chain
Cubase 15	VST3	B	Instrument + MIDI-out route (no VST3 MIDI-insert)
Cakewalk Sonar	VST3	B	Enable plugin MIDI-out, MIDI track → instrument
Studio One 7	VST3	B	Track B **Instrument Input = LeadMuse** ⚠ real-time export
Fender Studio Pro 8	VST3	B	Track B **Instrument Input = LeadMuse** ⚠ real-time export
Digital Performer	VST3 / AU	B	LeadMuse instrument → MIDI out to a 2nd instrument track
UAD Luna Pro	AU	B	Instrument track MIDI-out → instrument track
Pro Tools	AAX ✗	C	Import the `.mid`, or a VST→AAX bridge (Chapter 15)

> **Note.** Model-B export caveat. In Studio One and Fender Studio, MIDI routed this way is dropped on an *offline* bounce, adding roughly 50–100 ms of latency to the export. Either export with **Real-Time Processing** turned on, or sidestep the whole issue with **Drag-MIDI** (Chapter 12).

> **Note.** Cubase won't list a JUCE VST3 MIDI effect as a MIDI Insert. That's a limitation in how Cubase and JUCE-built plugins interact, not a setting you're missing — no category tweak fixes it. Use Model B

> instead: load LeadMuse as an instrument, enable its MIDI output, and route that into the real instrument. Cubase's own Chord Track pairs nicely with this setup for feeding Follow mode.

> **Note.** Chord Tracks and Follow mode don't behave the same way in every host. Cubase's Chord Track routes as real MIDI directly into LeadMuse's Follow input — it just works. Studio One's and Fender Studio's Chord Track is different under the hood: it's a *Follow Chords* reference that re-pitches other tracks assigned to follow it, and it never emits a routable MIDI chord stream of its own. In those two hosts, either feed LeadMuse real chord MIDI directly (a clip, or you playing live on the routed track), or put your chords on a track with **Follow Chords** turned on and route *that* track into LeadMuse. Song mode — LeadMuse's own grid — sidesteps this distinction entirely, since it never depends on a host's chord track at all.

Preview: hear the part before you route anything

Routing is worth learning, but sometimes you just want to hear what LeadMuse came up with — right now, before you've wired an instrument to it at all. That's what **Preview** is for. You'll find it just to the right of the range controls: a **Preview** button, and beside it a small level slider.

Figure 14.1 The **Preview** control, just to the right of the range controls: the Preview button and its level slider.

Arm Preview and press **Play** in your DAW. LeadMuse auditions the generated part through a built-in sound, locked to the transport exactly as the finished part would be — so what you hear is the real timing and feel, not an approximation. While Preview is on, the MIDI that would normally drive your routed instrument is held back, so you hear the built-in sound on its own rather than both at once. Switch Preview off and LeadMuse goes straight back to driving your own instrument, untouched — nothing about your routing changes. Preview just borrows the output for a moment so you can listen. Because it follows the transport, you won't hear anything until you press Play; that's by design, so a preview never plays over the top of a stopped session.

The small menu next to the button picks the preview guitar — **Acoustic**, **Electric (Clean)** or **Electric (Distorted)** — so you can audition a part with roughly the tone you have in mind. (As noted in Chapter 11, the preview sampler plays the notes but not the pitch-bend, so bends ride through your own routed instrument.)

> **Note.** The preview sound is an *optional* part of the installer. If you left it out, the Preview button and its slider are simply greyed out — everything else works exactly as before, and you route to your own instrument as usual. To switch Preview on later, run the installer again and tick **Preview instrument samples** (or, from the manual-install ZIP, follow the short readme in the `Preview` folder). Preview is only ever a quick audition aid; the sound you'll actually keep comes from the instrument you route to, or from the MIDI you drag out (Chapter 12).

Waveform 14 (Tracktion) — step by step

Waveform runs a left-to-right **plugin chain** on every track and passes MIDI straight down it, so LeadMuse and the instrument can live on one track together (Model A). The signal path is simply `chords → LeadMuse → instrument`.

1. **Add a track** — a normal one, nothing special.
2. **Drop LeadMuse (VST3)** into that track's plugin chain.
3. **Add an instrument to the right of LeadMuse**, in the same chain — Waveform's own Collective or 4OSC, or any VSTi you like. Order matters here: LeadMuse has to sit *before* the instrument (to its left) so its MIDI actually reaches it.
4. **Set your chords.** Song mode — LeadMuse's own grid — is the simplest option and needs no routing at all. For Follow mode instead, put a chord MIDI clip on that same track, ahead of LeadMuse in the chain, or just play chords live into the track.
5. **Press Play.** LeadMuse locks to Waveform's transport and plays the part through the instrument, in time.

> **Tip.** If nothing plays, drop an empty MIDI clip on the track spanning your arrangement — that alone is enough to keep the plugin chain actively processing.

Or skip the routing altogether: drag LeadMuse's part straight onto the instrument's track with **Drag-MIDI** (Chapter 12) and press Play. It's the simplest path in this host, same as everywhere else.

Cubase 15 — step by step

Cubase treats LeadMuse as an **instrument**, never a MIDI insert, so LeadMuse and the real instrument live on **separate tracks** — LeadMuse generates the MIDI, and the instrument on a second track turns it into sound. The one thing to know going in is that no amount of rescanning will make LeadMuse appear in a track's MIDI-insert slots (see the note above); load it in the rack instead and the routing falls into place.

> **Warning.** Load LeadMuse from **Studio ▸ VST Instruments** (F11), not as a plain Instrument track. Only a rack-loaded instrument exposes its MIDI output to other tracks; a plugin dropped straight onto an Instrument track keeps its MIDI to itself, and the relay in step 4 has nothing to point at.

1. **Instrument track** — insert your instrument (MODO Bass, a piano or organ VSTi, a guitar VSTi — whatever suits the part) on its own Instrument track, and play a note to confirm it makes sound. This is what you'll actually hear.
2. **LeadMuse in the rack** — open **Studio ▸ VST Instruments** (F11) and load LeadMuse into an empty slot. If Cubase offers to create a MIDI track for it, decline — you'll add the routing tracks by hand.
3. **Enable LeadMuse's MIDI output** — in the rack, click LeadMuse's **Activate Outputs** button (the MIDI-out icon in the slot's control strip) and switch the port on. **LeadMuse — MIDI Out** now exists as a routable source.
4. **Relay track** — add a MIDI track, set its **Input** to **LeadMuse — MIDI Out** and its **Output** to the Instrument track from step 1. Turn on that track's **Monitor** so MIDI passes through live, not just during playback.

5. **Feed chords (Follow mode only)** — add one more MIDI track with its **Output** set to LeadMuse, and put a chord clip there, play chords in live, or hand it Cubase's **Chord Track**. In **Song** mode — LeadMuse's own grid — skip this track entirely; nothing needs to reach LeadMuse's input.
6. Press **Play**. LeadMuse generates, the relay track carries the notes to your instrument, and the part plays in time with Cubase's transport.

Cubase is the one host where the **Chord Track** pays off directly: it emits real MIDI into LeadMuse's Follow input, so a progression on the Chord Track drives the part with nothing else to wire. And, as everywhere, you can skip the routing altogether — drag LeadMuse's part straight onto the instrument's track with **Drag-MIDI** (Chapter 12) and press Play.

Studio One 7 / Fender Studio Pro 8 — step by step

Fender Studio Pro is built on Studio One, so the two route LeadMuse the same way, and these steps cover both. Like Cubase, they treat LeadMuse as an **instrument**, not a MIDI insert — so LeadMuse and the real instrument live on **separate Instrument tracks**. The trick that makes it click is Studio One's **Instrument Input**: an instrument track can take its MIDI *from another instrument*, so you point your real instrument's input at LeadMuse and it plays whatever LeadMuse generates.

1. **Instrument track** — add an Instrument track and load your real instrument (a bass, piano/organ, or guitar VSTi — whatever suits the part). Play a note to confirm it makes sound; this is what you'll actually hear.
2. **LeadMuse track** — add a second Instrument track and load LeadMuse onto it. This one only generates MIDI; you won't listen to it directly.
3. **Point the instrument at LeadMuse** — select your real-instrument track from step 1 and open its **input** selector (the track's MIDI input, in the track inspector). Instead of a keyboard, choose **LeadMuse** as the input. That's the whole routing: the instrument now receives LeadMuse's output.
4. **Monitor on** — turn on that instrument track's **Monitor** (the speaker icon) so MIDI passes through live, not only during playback.
5. **Feed chords (Follow mode only)** — in **Song** mode, LeadMuse's own grid, skip this entirely. For **Follow** mode, put a chord clip on LeadMuse's track or play chords in live. Note that Studio One's and Fender Studio's **Chord Track** is a *Follow Chords* reference, not a routable MIDI stream (see the note earlier in this chapter), so feed LeadMuse real chord MIDI — a clip, live playing, or a track with **Follow Chords** turned on routed into LeadMuse.
6. Press **Play**. LeadMuse generates, your instrument track receives it through the Instrument Input, and the part plays in time with the transport.

> **Warning.** Export in real time. Both hosts drop this routed MIDI on an *offline* bounce (it arrives ~50–100 ms late, or silent), so turn on **Real-Time Processing** in the export dialog — or sidestep the whole thing with **Drag-MIDI** (Chapter 12): drop LeadMuse's part straight onto the instrument's track and bounce normally.

Cakewalk Sonar — step by step

In Cakewalk, LeadMuse and the instrument are always **two separate tracks** — LeadMuse emits MIDI, and the instrument is what actually makes sound.

> **Warning.** Don't drop the instrument into LeadMuse's own **FX bin**. That bin exists for *audio* inserts, and LeadMuse's audio output is silent by design — an instrument placed there gets nothing to work with. In the same vein, dragging MIDI onto LeadMuse's own track feeds it *into* LeadMuse, not into the instrument you actually want to hear.

1. **Instrument track** — insert your instrument as its own Instrument track first, and confirm it makes sound when you play a note on it directly.
2. **LeadMuse track** — insert LeadMuse as an instrument. In **Views ▸ Synth Rack**, right-click LeadMuse and choose **Enable MIDI Output**.
3. **MIDI relay track** — add a MIDI track, set its **Input** to **LeadMuse (MIDI Output)**, and its **Output** to the instrument.
4. Turn on that MIDI track's **Input Echo**, so MIDI actually passes through live rather than only on playback.
5. Press **Play**. In Song or Demo source, LeadMuse generates entirely on its own; in Follow source, feed chords to LeadMuse's own track — Sonar's Chord dock works well for this.

Or, once again, skip the routing entirely: drag LeadMuse's MIDI straight onto the *instrument's* track (not LeadMuse's own) with Drag-MIDI, and press Play.

15 Pro Tools

Pro Tools speaks **AAX only**, and LeadMuse doesn't have an AAX build yet — so it won't show up in Pro Tools' plugin list the way it would in almost any other DAW. That's not a bug to work around so much as a genuine gap, and I'd rather tell you that plainly than pretend otherwise. The good news is there are two workflows that cover Pro Tools today, and one of them needs nothing beyond LeadMuse itself.

Path 1 — import the MIDI

This is the most reliable path, and it doesn't cost you anything beyond what you already own.

1. Produce a `.mid` file of your part. **Drag-MIDI** (Chapter 12) or a plain Save both work, from LeadMuse running in any other DAW you own.
2. In Pro Tools, go to **File ▸ Import ▸ MIDI...** (or just drag the file straight in) and drop it onto an Instrument track that already has an instrument loaded.
3. That's it. Playback and mixdown are entirely native from here — no routing, no bridge plugin, nothing else to configure.

Because this path never touches LeadMuse at playback time, it's the one I'd reach for first: there's simply nothing to go wrong at mixdown.

Path 2 — a VST→AAX bridge

If you'd rather have LeadMuse actually running live inside Pro Tools — following chords in real time rather than working from a pre-rendered file — a bridge plugin makes that possible, at the cost of needing to own one.

1. Insert a bridge plugin on a Pro Tools Instrument track — **Blue Cat PatchWork (AAX)** or **DDMF Metaplugin** both work.
2. Inside the bridge, load **LeadMuse (VST3)**, then load an instrument (VSTi) alongside it.
3. Route LeadMuse's MIDI output into the instrument's MIDI input, inside the bridge.
4. Feed chords to the track. Both Follow mode and Song mode play normally through this setup, exactly as they would in a DAW that hosts LeadMuse natively.

A native AAX build — one that would show up directly in Pro Tools' own plugin list, no bridge required — is technically feasible, and the groundwork for it already exists in the build system. It isn't scheduled for a specific release, though: it depends on enrolling in Avid's developer and PACE signing programs, which is more of a standalone project than a feature flag, and it's being held for when there's clear demand from Pro Tools users rather than built speculatively. Until then, the two paths above are the whole story for Pro Tools, on both products.

16 Troubleshooting

Most problems with LeadMuse trace back to one of a handful of causes, and once you know the short list, a symptom that looked mysterious usually turns out to be a one-line fix. Here's the table to check first, followed by how to reach us for anything it doesn't cover.

Symptom	Fix
Plugin doesn't appear	Rescan plug-ins in your DAW. Confirm it actually installed to the system VST3 / Components folder (Chapter 2). On Mac, Logic and Luna specifically need the **AU** build, not the VST3.
No sound	LeadMuse makes no audio of its own — its **MIDI has to reach an instrument** somewhere in your chain. Check the routing pattern for your DAW (Chapter 14).
Part doesn't move / nothing plays	In **Song** mode, make sure you've actually got chords in the grid. In **Follow** mode, confirm chords are really arriving at LeadMuse's MIDI input — not just sent from your DAW, but landing on the right track.
Not in time	LeadMuse locks to your host's transport, not its own clock — press **Play** in the DAW itself, and check that **Host BPM** reads your actual tempo.
Drum-lock does nothing	Send kick MIDI on **channel 10** (notes 35 or 36) into LeadMuse, then raise the **Drum** slider (Chapter 11). Audio drums won't drive it — it only reacts to MIDI.
Bounce is silent or late (Studio One, Fender Studio Pro)	Export with **Real-Time Processing** turned on, or sidestep it entirely with **Drag-MIDI** (Chapter 12).
SmartScreen / Gatekeeper prompt	Release builds are code-signed, so this is a reputation check, not a sign anything's wrong — a brand-new release can still trip Windows' "not commonly downloaded" warning until it builds up a track record (**More info** → **Run anyway**), or need a first-time **right-click** → **Open** on macOS. See Chapter 2.
Shared song is missing a style	That section quietly falls back to the first style in the list rather than breaking. Ask whoever sent it for a **bundle** instead (Chapter 13) — a bundle carries every style it needs, so nothing has to fall back.

Report a problem or request a feature

Click **Report** in the plugin header to open a short form, right there in the plugin — no need to leave your DAW or dig up your version number. It asks for a **type** (Bug, Idea, or Question), a short title, and what happened, and it automatically attaches your LeadMuse version, your operating system, and the host you're running in, so you never have to type any of that yourself. If you'd rather not use the in-plugin form, the feedback page does the same job from a browser. Either way, it lands on the same feedback board, where other owners can see it and upvote it.

> **Warning.** Reporting a problem requires LeadMuse to be activated first — clicking **Report** before you've entered a trial or purchase code opens the activation screen instead of the report form. If you're troubleshooting during an unactivated trial, activate first (even the free trial counts) before you can send a report.

> **Note.** You'll be kept in the loop. When the status of something you submitted changes — Planned, In progress, Fixed, Declined, or No action required — you'll get an email at your account address, so you're not stuck refreshing the feedback board to find out.

17 Appendix & reference

This chapter is the reference material everything else in the manual points back to — file locations, the complete dial-by-dial breakdown of a style file, and the chord vocabulary LeadMuse reads. You won't read it start to finish so much as land on the one row you actually need.

File locations

Item	Path
VST3 (Windows)	`C:\Program Files\Common Files\VST3\LeadMuse.vst3`
VST3 (macOS)	`/Library/Audio/Plug-Ins/VST3/LeadMuse.vst3`
AU (macOS)	`/Library/Audio/Plug-Ins/Components/LeadMuse.component`
User styles	`Documents/LeadMuse/Styles/*.lmstyle`
User songs	`Documents/LeadMuse/Songs/*.lmsong`

`.lmstyle` keys — the full dial reference

Every dial a style can set. A LeadMuse style describes *where the lead sits and how it sings* — its register, its note length, and its bend, vibrato and run character. Seven are live controls in the plugin: the knobs **Density**, **Swing**, **Humanize**, **Ring**, **Bend**, **Vibrato** and **Runs** (all marked ●). The register block and the three lead-expression keys are what the lead engine actually reads; a few general keys (density, swing, humanize, velocity, timing) are shared across the Muse engines and behave the same here. The note-choice-weight, pianistic-gesture and strum keys that **Export Style** also writes are shared engine fields the lead generator ignores — leave them as they are. Unknown keys are ignored on import, so the format stays forward-compatible.

Key	Range · default	What it does
Identity		
`id`	text	Unique id a song uses to reference the style
`name`	text	Label shown in the Style dropdown
Register (MIDI note numbers)		
`rhLow`	0–127 · 55	Lowest note the lead reaches
`rhCenter`	0–127 · 64	Home register the line gravitates to, and where it starts
`rhHigh`	0–127 · 79	Highest note the lead reaches (the Solo role climbs a little above it)
Rhythm & density		
`density` ●	0–1.5 · 1.0	Baseline busyness (also the Density knob)

`swing` ●	0–0.5 · 0	Delays the off-beat within a beat (also the Swing knob)
`restProb`	0–1 · 0.05	Chance any non-anchor onset is dropped (space)
`ghostProb` ●	0–1 · 0	Chance a note becomes a quiet ghost note (the Ghost knob)
Feel (loaded into the knobs when the style is picked)		
`humanize` ●	0–1 · 0.6	Overall timing + velocity looseness (Humanize knob)
Lead expression		
`leadBend` ●	0–1 · 0.20	How often a held note bends up a whole step and settles back — a real pitch-bend (the Bend knob)
`leadVibrato` ●	0–1 · 0.40	How often a held note gets a CC1 vibrato swell (the Vibrato knob)
`leadRun` ●	0–1 · 0.45	Fast-run tendency — sparse phrasing vs. longer, busier licks (the Runs knob)
Shared engine keys (written by Export; ignored by the lead)		
`wRoot`	≥0 · 3.0	Pull toward the chord root
`wFifth`	≥0 · 1.5	Pull toward the fifth
`wOctave`	≥0 · 1.0	Pull toward the octave
`wThird`	≥0 · 0.8	Pull toward the third
`wSeventh`	≥0 · 0.6	Pull toward the seventh
`wApproach`	≥0 · 1.0	Chromatic/scale approach into the next chord
`wChromatic`	≥0 · 0.4	Chromatic passing tone off the previous note
`wScaleWalk`	≥0 · 0.8	Step through the chord scale (walking)
`octaveJumpProb`	0–1 · 0.10	Chance of leaping an octave
`approachOnChangeProb`	0–1 · 0.50	Chance of aiming at the next root on a chord change
Articulation		
`baseVelocity`	1–127 · 92	Loudness center
`velRange`	≥0 · 14	± velocity humanisation
`staccato` ●	0–1 · 0.80	Note length within the beat, as a fraction of its slot (the Legato knob)
Micro-timing		
`timingJitter`	beats · 0.010	Random timing spread per note

`behindBeat`	beats · 0.000	Constant push; positive = laid back, negative = on top

Rhythm cells

LeadMuse has no rhythm cells and no strum pattern — a lead line's rhythm comes from the phrasing engine itself (Chapter 11), shaped live by the **Role**, **Density** and **Runs**. A style sets *where* the line sits and *how* it sings — the register, the note length, and the bend / vibrato / run character — and the engine builds the phrases from there, landing on chord tones and stepping through the scale between them.

Chord symbols LeadMuse understands

Roots `A` – `G` with `#` / `b`; qualities including `m`, `7`, `maj7`, `m7`, `m7b5`, `dim`, `aug`, `sus2` / `sus4`; extensions `9` / `11` / `13`; slash bass like `C/E`. In a plain-text song file, a chord can be positioned within its bar as `SYM@beat` — for example, `G@2`.

LeadMuse v1.1.1 · VST3, Audio Unit, CLAP & AAX Native · Windows & macOS. Styles model stylistic characteristics, not copyrighted transcriptions of any specific recording or performer.