AbleSet 2

by Leolabs

Made for Ableton Live 10 & 11
Also available at ableset.app/docs

Getting Started

Getting started using the app is simple. AbleSet lives as a little icon in your menu bar. Click on it to open the main window.

To open the AbleSet interface on your computer, click "Open AbleSet" in the menu bar window. Your browser will open with the local address of AbleSet on your computer. If you want to open AbleSet on your phone or tablet, make sure they are on the same network as your computer and open either the Host or Remote address in their browser.

AbleSet uses Ableton's locators (markers) to convert your session into a setlist. The easiest way to set up a setlist is to place a marker at the beginning of each track and give it the song's name. If you want to know more about what you can do with markers in AbleSet, check out the Cue Notation documentation.

AbleSet's Cue Notation

AbleSet uses Ableton's locators (short: cues) to determine where songs start and end. The simplest form of a cue contains just the song title and is located at the song's beginning. Using this cue will lead to that title appearing as a song in AbleSet.

To ensure that playback runs smoothly, you should always place cues on a beat, if possible on a downbeat. If you place the cue between two beats, Ableton's quantization might not work as expected.

To give AbleSet more information on what a marker means, you can use a more advanced cue notation:

Subcues

A subcue starts with a > followed by the name – for example *> Verse 2*. You can use it to divide a song into multiple sections. Those sections can then be triggered from AbleSet.

If you want your subcue to be quickly accessible within performance view, you can prefix it with >> – for example *>> Chorus*.

Sections can also be defined using MIDI clips on a Sections track starting with Live 11. This causes less visual clutter in your arrangement view. You can learn more about it here.

Cue Descriptions

Everything written in curly brackets is interpreted as a description. You can use this feature to add information like the song's key, tempo, and length.

Example: *Follow Night {Tuning: C, Capo 1}*

You can also add descriptions to subcues.

Stop Markers

There are two ways to get AbleSet to automatically stop playback after a song. If you place a cue named *STOP* or *AUTOSTOP* in your set, AbleSet will automatically stop playback there.

If you place all songs directly next to each other, you can add a dot to the beginning of the song cue that follows the song after which you want playback to stop (for example: *. Song*). AbleSet will stop playback when it reaches this cue.

Normally, AbleSet will either jump to the next song's cue or stay at the stop cue upon reaching it depending on the "Autojump to the Next Song" setting, but you can override this behavior by adding either a *+JUMP* or *+STAY* flag to the stop cue – for example *STOP +JUMP* tells AbleSet to automatically jump to the next song's cue when the stop marker is reached.

Pause Markers

Using pause markers, you can tell AbleSet to pause playback at a certain section. To do that, you can add a *+PAUSE* flag to the cue – for example *> Chorus +PAUSE*.

Song End Markers

If you want to mark the end of a song but don't want playback to stop, you can place a cue named *SONG END* where your song ends. When this cue is reached, AbleSet will directly jump to the next song in the setlist if there is one.

Loop Markers

If you want to loop a section of your song, you can do so by adding *+LOOP* to the end of the cue where you want the loop to start. AbleSet will then loop from this cue until the next cue until you skip to the next marker in the web app. For example: *> Chorus +LOOP*.

If you want AbleSet to always finish a loop before jumping to the next cue, you can use *+LOOPFULL* instead of *+LOOP*.

Song Durations

AbleSet automatically measures the duration of songs, but if you want to set the duration of a song manually, you can do so by adding it to the end of the cue in square brackets. For example, *Follow Night [3:20]* will set the song's duration to 3:20 minutes.

This can be helpful for songs with variable tempo for which AbleSet can't automatically determine the duration.

Colors

To color-code cues, you can add a color name to the cue in square brackets – e.g. *Galvanize [blue]*. By default, subcues use the same color as the song they're assigned to, but you can override the color by adding a color name to the subcue – e.g. *> Verse [red]*.

The following colors are supported:

Gray, Red, Orange, Amber, Yellow, Lime, Green, Emerald, Teal, Cyan, Sky, Blue, Indigo, Violet, Purple, Fuchsia, Pink, and Rose

Ignored Cues

If you want AbleSet to ignore a cue, just prefix it with a *. This might be interesting for making notes in your live session, e.g. ** Check automation.*

AbleSet also ignores Clyphx cues, like *[STOP] SETSTOP; MIDI NOTE 3 1 127;* so they don't clutter up your setlist.

Controlling AbleSet with your MIDI Controller

If you prefer controlling your setlist via MIDI instead of a phone or tablet, you can use AbleSet's Max for Live device. You can find it by going to AbleSet's settings menu and clicking on *Get AbleSet Controller*.

To use this device, just drag it on any audio or MIDI track in your Ableton session. The controller will connect to AbleSet automatically. You can then map any of the functions to your MIDI controller as you're used to.

The Play, Pause, Next Song, and Prev. Song buttons are self-explanatory, but the Play Cued Song button has a special behavior. It only plays the next song after playback has been stopped by a *STOP* marker, or the current song if playback is stopped and the playhead is on the start marker of a song. This is a safety measure that prevents accidental skips to the next song while the current song is still playing. In the most minimal setup, you only need to map this button to your MIDI controller.

Connecting to AbleSet From Other Devices

AbleSet hosts its own server which you can access from any other device's browser as long as it is on the same network as your computer. This doesn't require an internet connection, so you can set up your own local network on stage.

There are a few different ways to create a network, and some might suit your setup more than others.

Connect an iPhone or iPad to your Mac

If you don't use AbleNet and just want to connect to AbleSet from an iPhone or iPad with a wired connection and without any extra dongles, this is the easiest way.

First, connect your iPhone or iPad to your Mac via USB. Then, open your Sharing preferences and go to "Internet Sharing" in the sidebar.

Make sure that "Share your connection from" is set to Wi-Fi. Your WiFi doesn't need to be connected to any network for this to work, but for some reason choosing Wi-Fi here works best.

In the list of "To computers using", check your iPhone's or iPad's connection – it should be called "iPhone USB" or "iPad USB".

Then check the checkbox next to "Internet Sharing" to turn sharing on.

You can now connect to AbleSet using the .local domain or the remote IP address that is shown in AbleSet's menu bar window.

Create a Network Using a Router

If you want to connect to AbleSet from multiple devices, you can use a router. This is one of the most common setups for on-stage networks and any simple router with or without Wi-Fi can be used for this.

Follow your router's instruction manual to set it up and then connect both your computer and the devices you want to use to connect to AbleSet to it. Most routers will automatically assign IP addresses so you don't need to worry about that.

Once all devices are connected, you can access AbleSet using either the .local domain or the remote IP address shown in the app's menu bar window.

Although it is possible to connect to AbleSet using Wi-Fi, I'd recommend using a wired Ethernet connection on stage since there will likely be lots of wireless interference from microphones, IEM systems, and other devices.

There are Lightning/USB-C to Ethernet dongles available if you want to connect a phone or tablet to your network with a wired connection.

Create a Network Using a Switch

This is similar to setting up a network using a router, except that switches often don't assign IP addresses to your devices automatically. You can manually assign IP addresses to your devices by going to the device's network settings and setting the "Configure IPv4" option to "Manually", or similar depending the device you use.

When choosing an IP address, make sure that the first three parts match on all devices. For example, you could use the IP address 192.168.1.100 for your first device, 192.168.1.101 for the second, and so on.

Make sure that the subnet mask is set to 255.255.255.0 on all devices.

Connect Two Devices Directly Using Ethernet

Instead of using a switch, you can also connect two devices directly using Ethernet. This is useful if you only have two devices and don't want to use a router or switch.

If you connect two Apple devices – iPhone/iPad/Mac – directly using Ethernet, they should automatically assign IP addresses, but I'd still recommend using the steps described in the section above to assign static IP addresses to both devices.

Redundant Playback Setup with AbleNet

AbleNet allows you to control two or more computers in a redundant playback setup. AbleNet is decentralized, so any computer in the setup can crash or lose connection – as long as one computer is still online, AbleSet will continue working.

To setup AbleNet, open the same Ableton session and enable AbleNet in the settings menu of AbleSet on all computers. The computers will automatically connect to each other and can now be controlled simultaneously with AbleSet.

You can open AbleSet in the browser or on your mobile device using either computer's IP address. When the computer you're currently connected to goes offline, AbleSet will automatically redirect you to the next available computer.

To connect two computers to the same output channels, you could use iConnectivity's PlayAUDIO12. This audio interface allows you to connect two computers and automatically switches to the backup computer when it detects that the main one isn't outputting audio anymore.

Automatic Drift Correction

Sometimes, your instances of Live might not start playback exactly at the same time, either because of network or disk latency. AbleNet offers an Automatic Drift Correction that takes care of nudging your computers back into sync.

To do this, AbleSet uses Live's Phase Nudge buttons to slightly slow down or speed up playback until all computers are in sync. You'll see the buttons flash while this happens.

In most cases, you might only want to enable this setting on your backup computer so that it can sync to your main computer and your main computer doesn't speed up or slow down playback, even if just slightly, to get in sync with the backup one.

For Automatic Drift Correction to work correctly, all audio clips have to be warped so they can follow the slight tempo changes caused by the Phase Nudge adjustments. Unwarped audio clips will continue playing and get out of sync with Live's metronome.

I'd recommend using the Texture or Beats warp mode for audio clips as they both don't alter the audio output at all while playing at normal speed.

Custom IP Addresses

AbleNet uses Apple's Bonjour protocol to automatically detect other instances of AbleSet on the network. This will work in most cases, but if it doesn't, you can manually specify a list of IP addresses that AbleSet should connect to.

To do that, go to AbleSet's settings menu and click on AbleNet -> Custom IP Addresses.... In a setup with two computers, you can enter the IP address of the other computer respectively in the window that opens.

Giving your computers static IP addresses simplifies this setup since you don't have to update this setting every time your computers get automatically assigned a new one. You can read more about this on the Network docs.

If AbleSet isn't running on port 3000, you can add the custom port to the IP address, e.g. 192.168.198.20:3005.

Sync Playback Now

In case one of your computers or Ableton crashes, you can use the "Sync Playback Now" option to synchronize its playhead position with the one of your other computer and start playback so that both sessions are playing in sync again. This usually takes just a few seconds.

Tips for Using AbleNet

- Ideally, you should connect all computers via Ethernet. That way, they're not susceptible to wireless interference which could cause latency issues.
- Make sure that you're loading the exact same set on all computers. Otherwise, the computers might behave differently when jumping between songs.

Lyrics in AbleSet

Since version 2, AbleSet supports displaying synchronized lyrics with your songs. Each track that contains the *+LYRICS* flag is recognized as a lyrics track. To get started, create a MIDI track called Vocals *+LYRICS* that contains one clip per lyrics line. AbleSet will automatically create a synchronized lyrics view based on this.

Creating these lyrics clips manually can be a chore, so I've created a tool that allows you to paste your lyrics, synchronize them to your backing track, and download them as a Live project containing a track with your synchronized lyrics: ableset.app/tools/lyrics

Multiple Lyrics Tracks

AbleSet supports multiple lyrics tracks, so you could create a *Lead Vocals +LYRICS* track for your lead singer, a *Backing Vocals +LYRICS* track for your backing singers, and a *Chords +LYRICS* track for other musicians.

If multiple tracks are available, each device can choose which one they'd like to see on the web app.

Styling

The lyrics view supports basic Markdown for making parts of your lyrics bold or italic. To make a part of your lyrics italic, wrap it in single asterisks, e.g. *italic line*. To make a part of your lyrics bold, wrap it in double asterisks, e.g. **bold line**.

To give a line a color, you can add it in square brackets to the beginning of the line, e.g. *[blue] Why We Are*.

If you want all lines of a lyrics track to have the same color, you can add it to the track name in square brackets, e.g. *Vocals +LYRICS [blue]*.

The text size can be changed using *[large]*, *[small]*, or *[tiny]*, and the alignment can be changed using *[left]* or *[center]*.

Other available styling options include [nofade] to disable fading inactive lines, [nozoom] to disable zooming out inactive lines, and [linemarker] to show a small trigangular marker next to the current line.

You can combine these attributes, e.g: *Vocals +LYRICS [blue] [small] [left]*

Chords

To add chords to your lyrics, you can use the ChordPro syntax. As soon as one line contains chords, the entire song's lyrics track is recognized as a lyrics&chords track which changes the layout of lyrics, automatically aligning all lines to the left for better readability. For example:

```
[F] I know you tried so many ti[Am]mes
To be [G]who you used to be
[F] I know you tried so many ti[Am]mes [G]
```

Gets turned into:

```
F                            Am
  I know you tried so many times
      G
To be who you used to be
F                            Am  G
  I know you tried so many times
```

These chords can be transposed on a per-song level. Just add the number of semitones you'd like to transpose to the song's locator, e.g. *Follow Night +2* or *Follow Night -3*.

Using Multi-File Projects

AbleSet's multi-file project feature allows you to split your Ableton Live project into one project file per song. This is especially useful if you have a lot of songs and handling them all in one session either gets too messy or you notice Live struggling to keep up.

To enable the multi-file project mode, go to AbleSet's settings menu and click on *Multi-File Project -> Choose Project Folder...*

You can then select a folder containing all songs that should be loaded into the setlist. This could be a Live project folder containing multiple project files, or a normal folder containing multiple Live projects – the folder structure doesn't really matter.

The filename of your songs can contain cue notation like notes, colors, and the song duration – for example: *Why We Are {some notes} [blue] [3.42].als*

Since file systems don't support the : character in filenames, it can be replaced with a . for durations, as shown in the example above.

Each project file needs to contain at least one marker at the start of the song and one

STOP marker at the end of the song.

Options

The Save Changes Before Loading Project option allows you to decide whether the current project file should be saved when loading a new song. By default, this is enabled to prevent data loss, but if you're sure you don't want any changes to be saved, you can disable this to speed up switching between songs.

The ableset.json File

This file is stored in the folder you selected as the multi-file project root and contains information about the current setlist, e.g. the order of songs, the setlist name, etc.

Caveats

Since Live projects can take a little while to load, jumping from one song to another is possible when playback is stopped. As such, *SONG END* markers will be interpreted as *STOP* markers.

One project file can currently only contain one song. Since AbleSet doesn't know about the contents of a project file until it's opened in Live, it has to treat each file as one song.

Using a MIDI Track to Define Song Sections

Since version 1.9.0, you can use a MIDI track in Live 11 to define song sections. This cleans up the session since you have to use fewer markers.

To get started, create a MIDI track called "Sections" and add empty MIDI clips to it where the song sections are. AbleSet will recognize these clips as section markers and add them to your song. The clip duration doesn't matter – each section ends where the next one begins.

You can use the cue notation syntax to add comments and set a color for a marker.

If you like, you can combine the use of section cues and clips in your songs. If there's a cue and a section clip at the same time, AbleSet will display both.

Due to some limitations with Ableton Live's API, sections defined by clips can only be jumped to when Live isn't currently playing.

Even with these limitations, section clips are a great way for defining sections that you want to jump to during rehearsals – combined with the count-in function, this is a great way to practice certain sections of a song.

Using AbleSet's Count-In Feature

AbleSet has a Count-In feature which allows you to start playback of sections a few bars before the section starts. This is helpful for practice situations where you want to work on a specific section. You can enable this feature by going to AbleSet's settings menu and clicking on *Count-In -> Enable Count-In*.

The menu allows you to choose between a 1-bar, 2-bar, or 4-bar count-in. You can also choose to solo the click track during the count-in. AbleSet will look for the first track or group that contains "click" as the click track.

For count-in to work reliably in versions of Live older than 11.1.5, make sure that you set Live's grid size to a fixed "1 bar" or lower. Otherwise, AbleSet will not be able to correctly position the playhead, and you'll see a warning message.

Exporting and Importing Setlists

AbleSet allows you to save setlists for later use. This might be useful if you often switch between a few setlists or to make a backup of your setlist. To export your setlist, click the edit button and then the download button on AbleSet's setlist page.

To import a setlist, click the edit button and then drag the saved file into the page. In addition to AbleSet's own file format, the importer supports Strange Electronic's SETLIST file format, so you can easily import those setlists as well.