

DSX Hack



DSX HACK

User Documentation



Table of Contents

Table of Contents.....	2
Overview.....	4
History.....	5
System Requirements.....	7
Installation.....	8
Uninstallation.....	10
Authorization & Initial Startup.....	11
Audio Settings.....	13
Usage.....	14
Overview.....	14
Mapping MIDI channels to DSX Tracks.....	16
.....	17
Importing the MIDI File.....	18
Adjusting DSX Settings.....	19
Leader Duration.....	19
Tempo.....	19
Sequence Length.....	19
Programming Merges.....	22
Making the DSX Audio.....	24
Saving and/or Playing back the DSX Audio.....	25
Documentation & Help.....	26
About Menu.....	27
Demo Limitations.....	28

Acknowledgments and Thanks.....	29
Contact & Support.....	31
Copyright.....	31
Version History.....	31

Overview

DSX Hack is a software application that allows you to program the Oberheim DSX sequencer in an entirely new way, enabling you to use the sequencer without an accompanying Oberheim keyboard, which the DSX has required up until now. The application takes a MIDI file as input, and converts it into a WAV audio file that is played into the cassette input port of the DSX, to program its Tracks, Sequences and Merges.

DSX Hack is available for both Macintosh and Windows systems.

History

In 1997, I released the first version of [MC-202 Hack](#), a program that allowed users to convert MIDI sequences into WAV audio files that in turn programmed the Roland MC-202's sequencer. This was done by reverse-engineering the original cassette tape save format. I subsequently got many requests to do the same thing for other pre-MIDI sequencers, and in 2011 created [MC-4 Hack](#), doing the same thing for the Roland MC-4 sequencer. And so following the pattern, it's now time for the Oberheim DSX Hack (again at the prompting of several people from around the world)!

This latest creation included several challenges, among them that the Oberheim cassette format didn't resemble the two previous Roland versions very much at all. On the other hand, I was fortunate enough to get an incredible head start from one of the people behind the creation of the DSX, Marcus Ryle. I had originally contacted [Tom Oberheim](#) himself, who told me that all of the code that went into the DSX was Marcus's creation. Turns out that some 30 years later, Marcus is now the CEO of [Line 6](#), but nonetheless he miraculously responded to my email about the project and was incredibly gracious, even going so far as to review his old Z80 machine code (that Paul White of [Electrongate](#) generously contributed) to give me an accounting of how it created the sequencer tape saves. It was incomplete, and I encountered a few inconsistencies in his review that I had to work through, but it certainly made this whole endeavor that much more possible. For more background on Marcus and the DSX, there's a great [interview with him on KVR Audio](#), that includes his memories of creating the DSX, as well as his early conceptual line drawings of the DSX.

Creating DSX Hack proved difficult on another level as well; I didn't originally own a DSX

DSX Hack

myself! So all of the initial reverse engineering of the tape save format and the testing of early attempts was only possible with the help of some extremely helpful and dedicated beta testers, to whom I'm extremely grateful (see Acknowledgements section for their names). I did eventually pick up a DSX myself for more rapid testing, though I still don't have an Oberheim keyboard for testing...

So all that said - I'm happy to finally deliver this latest in a series of hardware sequencer "hacks" – Enjoy!

System Requirements

Generally, you will need to be able to get audio output from your computer to the DSX's tape input, so you will need the appropriate cable to do so.



Macintosh

- OS X 10.11.6 – 11.x (Big Sur compatible, runs on M1 chip via Rosetta 2)



Windows

- Windows 7, 8 or 10, 64-bit only



Oberheim DSX

- **OS 3.0 or greater; DSX Hack WILL NOT WORK with DSX OS 2.x or 1.x!** You can check to see what OS you have by pressing the 3, 5 and 7 keys down on the DSX at the same time; the version of your OS will be displayed on the LED. If that doesn't work, see what the LED displays when turning the DSX on without an Oberheim keyboard connected; if it shows “Connect Synth”, then you have a 2.x OS or lower. If it shows “! Oberheim DSX !” then you have a 3.x machine.

Installation



Macintosh

The Macintosh version is supplied as a Macintosh disk image file (.dmg). Just double click the downloaded file, and you will see a new window appear on your desktop. Drag the DSX Hack application icon to the Applications window icon that you see – this will install the DSX Hack application on your computer:



After the file has copied, you can unmount the disk image, and delete the .dmg file (or keep it somewhere safe if you'd like, for future installations).



Windows

The Windows version is supplied as a compressed .zip file. Unzip the file, and you will be left with a file called DSX Hack Installer.exe (or DSX Hack Demo Installer.exe if installing the demo version). Double click this file to run the installer, and follow its instructions:



After installation, you will find a shortcut available to launch the DSX Hack application in the Windows Start menu, in addition to uninstallation and documentation shortcuts. You can keep the DSX Hack Installer.exe file somewhere safe if you'd like, for future installations.

Uninstallation



Macintosh – Just drag the DSX Hack application out of the Applications folder to the Trash.

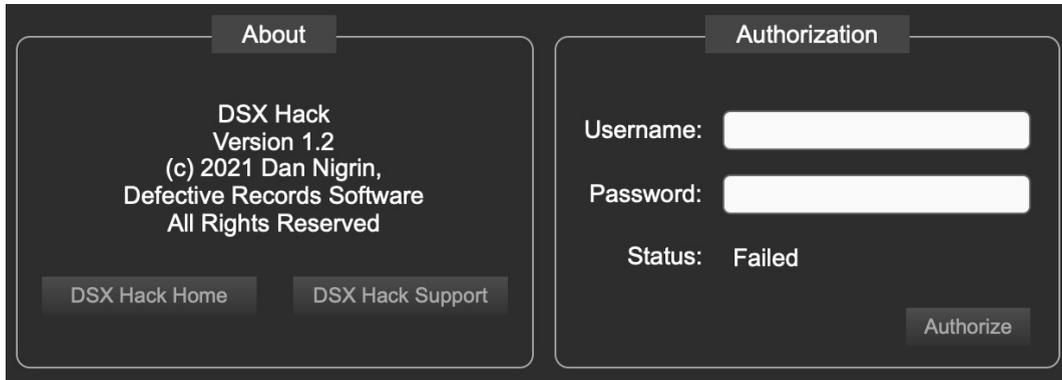


Windows – Use either the Windows Add/Remove Programs control panel to remove the application, or use the Uninstall shortcut in the Windows Start menu.

Authorization & Initial Startup

If you're trying the demo version of DSX Hack, you can start to use it immediately after installing it.

If you are installing the full version of DSX Hack, you'll need to authorize it before being able to use it. Start the application, and you'll find yourself at the About & Authorization screen:



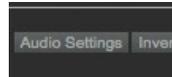
Enter your Username and Password, and then click the Authorize button. Be careful, cApiTaliZation MaTTeRs! If successful, the Status field will change to “Registered”, and you will see a “Use the App!” button, which when clicked will allow you to begin to use the software. If not successful, you will be left at the Authorization screen to try again.

DSX Hack

To verify your authorization status, or to remind yourself of your Username and Password, you can select the About DSX Hack menu item (or click the DSX Hack logo at the top left of the app) to be brought back to the About and Authorization screen at any time.

Audio Settings

You can configure the app's audio settings by clicking the Audio Settings button on the main screen.



This window allows you to turn audio processing on and off, select the audio driver used, configure the audio output device, and adjust settings that device. Some of the configuration options will depend on which driver and devices are selected, as well as whether you are working on a Macintosh or a Windows machine. You can also turn audio processing on and off by clicking the power button in the top right of the app.



Audio processing must be On to be able to send audio to the DSX.

Usage

Overview

DSX Hack works by converting sequence information from a MIDI file into audio that is then played into the DSX's cassette input port to program its sequencer.

The DSX contains ten Sequences, numbered 0 – 9. Each Sequence in turn contains ten Tracks, numbered 0 – 9. Track 0 of each Sequence contains information about patches, splits, doubles, tempo and sequence end point; DSX Hack can currently only encode information about tempo and sequence end point into Track 0. Tracks 1 – 9 contain note information.

Each Track 1 – 9 can send note information to 8 Voices on an Oberheim keyboard, or to the 8 CV/Gate output pairs available on the back of the DSX.

Finally, the DSX has ten Merges numbered 0 – 9 available. A Merge is simply a listing of Sequences to play in order, with the opportunity to transpose those Sequences for each step of the Merge.

DSX Hack's main user interface allows you to work on Sequences or Merges (the topmost two tabs):



DSX Hack

... and for each, allows you to select which Sequence or Merge you are currently working on (sub-tabs 0 through 9):

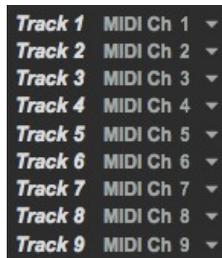


Using DSX Hack involves several simple steps:

- Choose settings for how MIDI channel information should be mapped to DSX Tracks and Voices, as well as whether or not each Track should use the CV outputs
- Import the MIDI File into your desired Sequence
- Adjust DSX settings as desired (e.g. Sequence Length, Tempo, Leader Duration, etc..)
- Program Merge(s) (if desired)
- Make the DSX Audio file
- Play the DSX audio file into the DSX (or Save it)

Mapping MIDI channels to DSX Tracks

Before importing a MIDI file(s), you need to specify how note information from the MIDI channels in the file should be mapped to Track and Voice information in the DSX Sequence. First, be sure you have the desired Sequence number sub-tab selected (from 0 – 9). Each DSX Track in the Sequence can be mapped to one MIDI channel (1 – 16); each MIDI channel can only be mapped to a single Track; choosing the same MIDI channel for another Track will deselect the MIDI channel from the previously assigned Track. By default, Tracks 1 – 9 are mapped to MIDI channels 1 – 9:

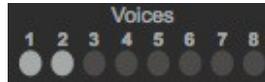


Track 1	MIDI Ch 1	▼
Track 2	MIDI Ch 2	▼
Track 3	MIDI Ch 3	▼
Track 4	MIDI Ch 4	▼
Track 5	MIDI Ch 5	▼
Track 6	MIDI Ch 6	▼
Track 7	MIDI Ch 7	▼
Track 8	MIDI Ch 8	▼
Track 9	MIDI Ch 9	▼

Each Track can be assigned to up to eight Voices on the DSX. These Voices map to those on a connected Oberheim keyboard, or if using the DSX's CV/Gate outputs, to CV/Gate pairs 1 – 8 on the DSX's back panel. To use the CV/Gate outputs, be sure to check the “To CV Outputs” box for each desired Track:



For each Track, choose the desired Voice(s) that should play the notes. If selecting multiple Voices for a Track and Send to CV Outputs is checked, notes will play sequentially through each selected CV/Gate pair.



Importing the MIDI File

Once you've specified the MIDI channel to DSX Track mapping, and selected the desired Voices for each Track, you're ready to import the MIDI file. ***Please note that adjusting the MIDI channel to DSX track mapping won't have any effect if done after the MIDI file has been imported.*** Drag and drop the MIDI file onto the main Sequences panel, or click the Select button to choose the MIDI file. Be sure you have the desired Sequence number sub-tab selected (from 0 – 9).



You can drag a different MIDI file to each of the Sequence sub-tabs, allowing you to send multiple Sequences at once to the DSX. MIDI file import works with either type 0 or type 1 MIDI files.

Because of note range constraints on the DSX, only MIDI notes 0 to 63 are permitted. Depending on if you categorize middle C as C4 (Roland format) or C3 (Yamaha format), this corresponds to C-1 to D#4, or C-2 to D#3, respectively. Notes above or below this range in the

source MIDI file will get scaled to fit into the 0 – 63 range.

Adjusting DSX Settings

Once you've loaded the MIDI file(s) into your desired Sequence(s), you can now further specify some additional settings. Some are global for all Sequences, and some are Sequence-specific:

Leader Duration

You can set the duration of the DSX cassette leader tone here, from 2 – 9 seconds. This is a global value for all Sequences.

Tempo

When you import the MIDI file, if it has a Tempo programmed into it, it will be automatically detected by DSX Hack, and defaulted into the Tempo field. You can adjust this to a different tempo if desired. If no tempo is specified in the MIDI file, a default value of 120 is selected. The range of allowable tempos is 25 – 250. Tempos are Sequence-specific.

Sequence Length

When you import a MIDI file, the respective Track Lengths are automatically shown next to each Track within a Sequence. The longest one of those Tracks becomes the default Sequence Length.

DSX Hack

The Sequence Length is important when using the DSX in Loop mode. It may be important for you to adjust the Sequence Length, as MIDI files do not include information about the absolute length of the original sequence in terms of beats and bars; they only include the information on the length and position of notes in the sequence. This means that a MIDI sequence will end at the exact time that the last note in the sequence has completed playing. Let's take for example the following sequence, created in Ableton Live:



The MIDI file exported from this sequence will only contain information about the notes included, and nothing about the overall length of the sequence that was originally intended (i.e. that it was a full bar long). The silence after the last note to “complete” the bar is not included in the MIDI file. So if you desire to loop this 1 bar sequence, it's advisable to manually adjust the Sequence Length value to reflect the length of the full sequence (e.g. 1 bar in this case).

The Sequence Length parameter is set in Bars, Beats and Units, where there are 480 Units/Beat.

Currently, DSX Hack is only configured to allow for 4/4 time signatures.

DSX Hack

The Sequence Length is dependent on Tempo; if you change the Tempo from that detected from within the MIDI file, the Sequence Length will change, and will not accurately depict the number of Bars/Beats/Units in your sequence. So best to adjust the Sequence Length with the original Tempo, and then if desired change the Tempo afterwards.

Programming Merges

Merges allow you to specify a “sequence of Sequences” for the DSX to play back. Each Merge is simply a list of Sequences to play back one after the other. Each step in the Merge can have a transpose amount assigned to it too. Transpose amounts are limited to positive amounts (higher pitches) – this is a limitation of the DSX. The transposition amount can be 0 to 63 steps. If the transposed notes exceed the topmost allowable note by the DSX, the notes will be scaled to fit within the allowable range.

To program your Merge, click the Merge tab towards the top of the DSX screen, and then select the desired Merge number (0 – 9). By default, you will not see any steps within the Merge; you can Insert steps above or below the currently selected step. You can also Delete individual steps, or Delete All steps.

Each step in the Merge represents a Sequence that will play back. Enter the desired Sequence number in the row, as well as any desired transposition amount.



You can also Save and Load Merges to your hard drive, for easy recall at a later time.

The step numbering code for Merges is a little buggy at the moment, don't worry if the first step is not #1, just ensure that the step numbers are in sequential order.

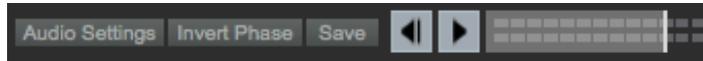
Making the DSX Audio

Once you have constructed your desired Sequence(s) and/or Merge(s), you are ready to make the DSX audio that will program the DSX. Simply press the Make DSX Audio button in the lower left corner. You will see a green progress bar briefly appear under the button – when it disappears, the DSX audio has been created.



Saving and/or Playing back the DSX Audio

Once you have completed creating the DSX audio, you can save the audio file or send it directly to the DSX.



To save the audio file, press the Save button, and you will be prompted with a file dialog where you can name the file. A .wav suffix will be placed at the end of your audio file by default.

You can also just send the audio directly to the DSX, by pressing the Play button. Adjust audio levels using the Volume slider. Audio will be transmitted per the audio settings specified in the Audio Settings. **Note that Audio must be turned On to send audio to the DSX!**

Follow the normal DSX owner's manual instructions for loading the audio into the DSX.

If you run into trouble with the DSX not reading the file correctly, it may be because the phase of the audio is not what the DSX is expecting; the DSX is exquisitely sensitive to the phase of the incoming audio. To correct for this, press the Invert Phase button, and try sending the audio to the DSX again. You can also experiment with the volume of the audio sent to the DSX; again, too loud or too soft incoming audio can result in the DSX not interpreting the audio correctly.

Documentation & Help

DSX Hack can launch its full documentation (this document) from within the application itself; just go to the Documentation menu, and choose “DSX Hack Documentation”. This will launch a PDF viewing application in which to view the documentation.

On Windows, you can also find a shortcut to the documentation in the DSX Hack Start menu shortcuts.

About Menu

The About DSX Hack menu item (in the DSX Hack menu on Macintosh, and in the Help menu on Windows) will replace the DSX Hack main screen with the About and Authorization screen. In addition to the Authorization information described above, it shows version and copyright information about the application, and has direct links to the DSX Hack home and support pages:



Demo Limitations

The demo version of DSX Hack has the following limitations compared with the full version:

- It only supports the inclusion of 4 notes on Track 1, and no notes on any other Tracks.
- It does not support Merges.

Acknowledgments and Thanks

This software would likely not have seen the light of day had it not been for the help of **Marcus Ryle**, CEO of Line 6, and his “decoding” of the original DSX firmware. In addition, **Paul White**, founder of Electrongate, provided the dump of the latest version of the DSX firmware which was critical as well. And thanks to **Tom Oberheim** himself for the tip to contact Marcus in the first place.

I would also like to acknowledge the following beta testers for their incredible assistance in helping me create and test this software: **Eric Frampton, Gino Giusti, Bruno Lhuissier, Cris Rodriguez** and **Doug Terrebonne**. They provided countless DSX cassette dumps to me, and helped me debug my early efforts in generating the audio files from the software. They also guided me with implementation decisions as the years ticked by. That's the other thing – they stuck by me for a LONG time (2.5 years!), since so many things kept getting in the way of me finishing. Thanks again guys!

I'd also like to thank **Martin Ward**, who's Perl-based Kansas City Standard (KCS) decoding software was invaluable in interpreting the raw cassette dumps. He also helped me in sorting out a specific problem in using his software to decode the DSX dumps. I also used Martin's code in my MC-4 Hack efforts.

DSX Hack was created using Cycling 74's Max (<http://www.cycling74.com>) software. In addition, it uses, with permission, some 3rd party external objects – I'm indebted to their creators for them:

DSX Hack

- `getpaths` (Mac), `creatfolder` (Mac, Windows); `getenv` (Windows), by **Jasch** (<http://www.jasch.ch/dl>)
- `min.environment` (Mac, Windows) by Tim Place
- `Lobjects` (Mac, Windows), by Peter Elsea
 - <http://artsites.ucsc.edu/EMS/music/research/lobjects.readme.html>
 - `Lobjects` are Copyright Peter Elsea and The Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
 - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
 - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
 - Neither the name of the University of California nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
 - THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Thanks to everyone at Cycling '74 and on the Cycling '74 forums (<http://www.cycling74.com/forums>) for their help with Max issues.

Contact & Support

Support Forum

<https://defectiverecords.com/forum/categories/dsx-hack>

Dan Nigrin

dan@defectiverecords.com

Copyright

DSX Hack is Copyright © 2016 - 2021 by Daniel Nigrin and Defective Records Software. All rights are reserved.

Version History

Version 1.21 (October 9, 2021)

- Fixed bug in which audio output to DSX was not properly recognized by DSX in some instances
- Built with Max version 8.2.0 (42e5454dc76).

Version 1.2 (March 7, 2021)

- 64-bit only
- Now Mac OS 11.x (Big Sur) compatible (including Mac M1 machines via Rosetta2)
- Enabled hi resolution mode on Mac

DSX Hack

- Basic Dark Mode support on Mac Mojave 10.14.x systems and greater
- Updated Defective logo
- Built with Max version 8.1.10 (flacba9) (64 bit)

Version 1.1 (November 12, 2016)

- Changed Send to CV Outputs to be per-Track, and not Global.
- Fixed bug where long Tracks (greater than 280 notes) were not being successfully sent to DSX.
- Built with Max version 7.3.1 (340a77a).

Version 1.0 (July 8, 2016)

- First version released, including demo version.
- Built with Max version 7.2.3 (46583fa).

Version 1.0 Betas 1 through 3 (March 18, 2016 – April 20, 2016)

- Beta testers included in testing.

Version 1.0 Alphas (December 2013 – March 2016)

- First versions for internal testing.