

# Algorithmic Arts'

## DataBin Data to Binary Converter

**Converts data files to binary MIDI range of 0-127**

[Display screen shot of DataBin](#)

### Overview:

DataBin reads just about any text based data file with numbers in any range. Once the source data file is read into the text window, you can pick whether to convert all numbers, or select a field and just convert the numbers in that field. It then converts the selected numbers to the legal MIDI range of 0-127. You save this conversion as a binary file that can be processed by MIDI algorithmic music software such as SoftStep.

To use DataBin, you read in a data file, trim it to size if you like, set whether to convert all numbers, numbers in a comma delimited field, or numbers in a character position column. Then you click the Convert button to convert the data and see it plotted in the graph area below the text window. If you are happy with the selection, you can then click on Save to save the data to a binary (.bin) file.

In order to save the binary file created by DataBin, you must register the software by purchasing a license key from [Algorithmic Arts](#). The unregistered version pops up a registration dialog when the program is first started. If you do not have the license key, just click on the dialog Exit button and the program will run normally, except that you can not save the binary data files.. To remove your registration key from the computer, right click on the Help button to pop up the registration dialog, leave both boxes blank and click on "Install," then "OK."

DataBin output can be imported to most algorithmic composing programs - it is not specific to SoftStep. However the File Sequencer module in SoftStep is specifically designed for this format, as it has no limit on the size of the data file other than normal computer resource limits; and any of SoftStep's other sequencers can load the data up to their set data size by using the sequencer's Fill button.

The reason DataBin makes binary files, but stops short of producing an instant MIDI file from the data, is to give the ability to use the data for other parameters in addition to the obvious one of pitch. With creative mapping of the data, including using it to affect other parameters such as Volume, Modulation, Pan, etc., you will discover data sonification can produce highly listenable music - music that has extra layers of depth added by the data you have chosen to sonify.

### Data Files:

Data files must be ASCII/ANSI text. If you can read it in NotePad it will be OK. Other than that, there are very few limitations. If you read a file that is not text, or that has no line feed characters in it to separate the lines, it will appear as unreadable garbage or as an empty page. This is your clue that the data file format is not useable by DataBin.

Some data files are very large - up to several megabytes. You can read and process these, but it will take extra time. Usually it will be better to break up the really large files into smaller ones, but there is no requirement to do so.

## Opening a Data File:

To load a data file, click on the **Open** button, and open the file with the standard file pop up window that will appear. DataBin treats the source data file as Read Only. That is, it never alters the source data file. If you make any changes, they only affect a temporary copy. The original files are **never** altered.

Some data files are stored in Unix format, but other than taking a little longer to read, this will be transparent to you, as the software automatically translates Unix files. The translated file is saved in a temporary file in the DataBin program directory ("temp.\$\$\$"). You should just ignore this file, as it will be changed every time a new Unix file is read.

## Data File Format:

DataBin can process three file formats. Free numbers, comma delimited numbers, and numbers arranged in columns. You will recognize these formats instantly when you load the file. Comma delimited format has lines of information separated by commas; column data shows up as neatly arranged columns of numbers and other data; and everything else can be considered free numbers.

The main difference between free numbers and the other two formats is that free numbers (selected with the "**All Nums**" option) processes everything that is a number, line by line, top to bottom. The other two options ("**Commas**" and "**Columns**") only process the numbers in the selected fields.

## Setting Comment Characters:

If the first character of a line is a comment character, as placed in the "**Comment**" box just under the **Open** button, the line is ignored and no numbers on that line will be processed. Since "#" is commonly used for comments, it is the default comment character, but you can type in any characters you like. As long as the first character of a line is one of the characters in the comment box, the line will be ignored.

## Selecting a Field to Process:

**All Nums** - the default selection - needs no further selection. You can immediately click on the **Convert** button, and all numbers in the file not on comment lines will be converted, and you will see a graph of the results. The other two selections, **Commas** and **Columns**, require you to select which field to process.

When you click on the **Commas** option, one of the two input boxes just below the options will become active, for you to type in the field number you want to process. Data fields are numbered left to right, starting with 1. You don't actually have to count which field you want - instead you can just put the cursor on one of the numbers, and then click on the Set button. The field number the cursor is positioned over is entered into the box. At this point you can click on the **Convert** button, and the numbers in that field in all the lines will be processed and you will see a graph of the results.

If the source data is in columns, click on the **Columns** option. Both of the input boxes below the options will become active. Type in the column number in the first box. This is the character number (starting with 1) that the column starts on. And type in the number of characters in the field in the second box. Or you can do it the easy way: Just highlight one of the numbers in the text, making sure to include all of the text area that make up the column, and click on the Set button as before. The input boxes will be filled in with the starting column of the highlighted text and the length of the highlight. At this point you can click on the **Convert** button to process and display the graph.

## Trimming the Data File:

To limit the amount of data to process, you can trim the data file. The **Del Above** button deletes all lines above the current line, which is the line the cursor is on. The **Del Below** button deletes all lines below the current line, and **Del Line** deletes the current line. Remember, the data file itself is never altered. These buttons just delete the lines to be processed. To restore deleted lines, simply click on the **Open** button and reload the file.

## How Data is Processed:

When you click on the **Convert** button, the numbers are collected from the data file and stored as an array of double precision floating point numbers. All processing up to the final conversion to MIDI integers is in double precision floating point, to maximize the accuracy of the data translation.

The numbers are then tested to see if any of them are negative. If they are, an offset is added to all numbers to make them all positive. The value of the offset is the absolute value of the smallest (most negative) number, so this least value becomes 0, and all other values are adjusted upward.

If there were no negative numbers, if the **Trim** check box is checked (the default), then the value of the lowest positive number is subtracted from all numbers, making the lowest number 0, with all other values adjusted upward.

If the **Normalize** check box is checked (the default), the highest value number is divided into 127, then all numbers are multiplied by this value. This normalizes all numbers to the legal MIDI range of 0-127. That is, if the number range was higher than 127, the numbers are compressed, if the range was less than 127, they are expanded. The only time you would not normalize values is if the number range is below 127, and you want to use the unmodified values.

Finally, all numbers are limit tested to insure none are below 0 or greater than 127. Numbers below 0 are set to 0, numbers greater than 127 are set to 127; after this is done, the numbers are converted from double precision floating point to byte integers, and they are stored in a byte array ready to be saved in a .bin file. Click on the **Save** button for the save file pop up window, to name and save the file.

## Mini Tutorial:

There is an example data file included with this program, that we will use for this mini tutorial. It is the NASDAQ daily averages for several stock categories over the first half of year 2000. This file came from the [NASDAQ website](#), where you can download daily averages from 1971 to the present.

Start by running DataBin, and opening the file "**Nasdaq2000.txt**" from the program directory. You will see a very dense page of text, and if you look closely you will see that the fields are separated by commas.

Notice the top line is a label line. It has the same fields, but they contain text:

```
"Date" , "N100" , "N100old" , "Financial100" , "Composite" , etc...
```

Then the lines following have the actual data:

```
1/3/00 0:00:00 , 3790.54 , 7581.08 , 1749.09 , 4131.15 , etc...
```

Since we don't want to process the label line, it can be ignored by typing the quote (") character into the **Comment** box. The quotation mark ("Date"... ) is the first character of the line, so even though it wasn't intended as a comment character, it works fine as one anyway.

Click on the **Comma** option box. The first of the two text entry boxes below becomes active. This box gets the field number of the one you want to capture. The easiest way to do this is to go to the top line and click on the name of the field., then click on the **Set** button. Then click on the **Convert** button to convert the data and see it displayed as a graph below the text window. Try this a few times with different fields.

## Making Music With Data:

The file produced by DataBin is a byte integer file, not a MIDI file. In order to hear it, or to turn it into a MIDI file, you have to import it into a MIDI composing program that can read binary files. Many can do this, but since this is Algorithmic Arts, we will use our own SoftStep. If you do not already have SoftStep, you can download it for free at the Algorithmic Arts web site: <http://algoart.com>

SoftStep v1.3, Release 9 and later have example patches that play DataBin files instantly. If your SoftStep is an earlier release, you can download these from the User File area in the Algorithmic Arts web site.

So, assuming you have SoftStep installed, the first step is to run SoftStep, and load the example patch called "**DataPlay**." This is a very simple patch that simply plays the data file as pitches. It is preloaded with data, but you can use the Get button on the file module to load in any of the .bin files you have created, and you will instantly hear them. Use the Tool Bar Compose button to create a MIDI file. The green **Info** button tells you exactly the steps to do this.

Other pre existing patches you can try out, are: "Nasdaq" and "StarData." As in the previous patch you can load in your own .bin files you have created with DataBin.

"Date"	"N100"	"N100old"	"Financial100"	"Composite"	"Composite_Hi"	"Composite_Low"
1/3/00 0:00:00	3790.54	7581.08	1749.09	4131.15	4192.19	3989.71
1/4/00 0:00:00	3546.19	7092.38	1688.34	3901.69	4131.15	3898.23
1/5/00 0:00:00	3507.31	7014.62	1664.15	3877.54	3924.21	3734.87
1/6/00 0:00:00	3340.80	6681.60	1701.98	3727.13	3877.54	3715.62
1/7/00 0:00:00	3529.59	7059.18	1718.46	3882.62	3882.67	3711.09
1/10/00 0:00:00	3717.40	7434.80	1694.71	4049.67	4068.40	3882.62
1/11/00 0:00:00	3544.35	7088.70	1658.01	3921.19	4066.66	3904.82
1/12/00 0:00:00	3478.13	6956.26	1651.97	3850.02	3950.98	3834.53
1/13/00 0:00:00	3612.07	7224.14	1710.73	3957.21	3957.47	3850.02
1/14/00 0:00:00	3704.73	7409.46	1715.75	4064.27	4091.95	3957.21
1/18/00 0:00:00	3757.77	7515.54	1693.68	4130.81	4148.00	4053.21
1/19/00 0:00:00	3790.88	7581.76	1677.96	4151.29	4164.65	4084.73
1/20/00 0:00:00	3841.73	7683.46	1636.19	4189.51	4227.35	4143.61
1/21/00 0:00:00	3849.95	7699.90	1629.02	4235.40	4238.00	4168.30
1/24/00 0:00:00	3660.95	7321.90	1597.20	4096.08	4303.15	4095.86
1/25/00 0:00:00	3759.11	7518.22	1621.23	4167.41	4167.63	4028.51
1/26/00 0:00:00	3621.20	7242.40	1644.40	4069.91	4174.72	4069.91
1/27/00 0:00:00	3593.14	7186.28	1643.06	4039.56	4140.09	3973.59
1/28/00 0:00:00	3446.13	6892.26	1613.88	3887.07	4048.31	3856.23
1/31/00 0:00:00	3570.05	7140.10	1660.75	3940.35	3940.46	3748.03
2/1/00 0:00:00	3701.78	7403.56	1656.51	4051.98	4053.16	3911.84

Open

Comment #

Del Above

Del Line

Del Below

Trim

Normalize

All Nums

Comma

Columns

5

Set

Convert

Num Lines 167

Num Values 166

Save

Help

Exit

