Voith’s CODE – Blackbox Buddy User’s Guide

Last updated: March 10th, 2022 for version 1.0.0.550++

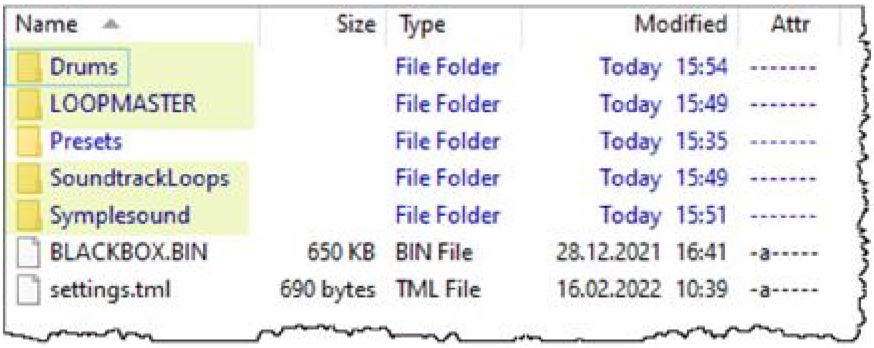
Homepage: https://www.vcode.no/bbb
Contact: Bob Voith, bob@vcode.no

This document describes the command-line tool for Windows named “Blackbox Buddy” or BBB for short.

Why BBB?

“I wanted to be able to rename, and move around my samples without any Blackbox-presets, loose their references to the sample files”.

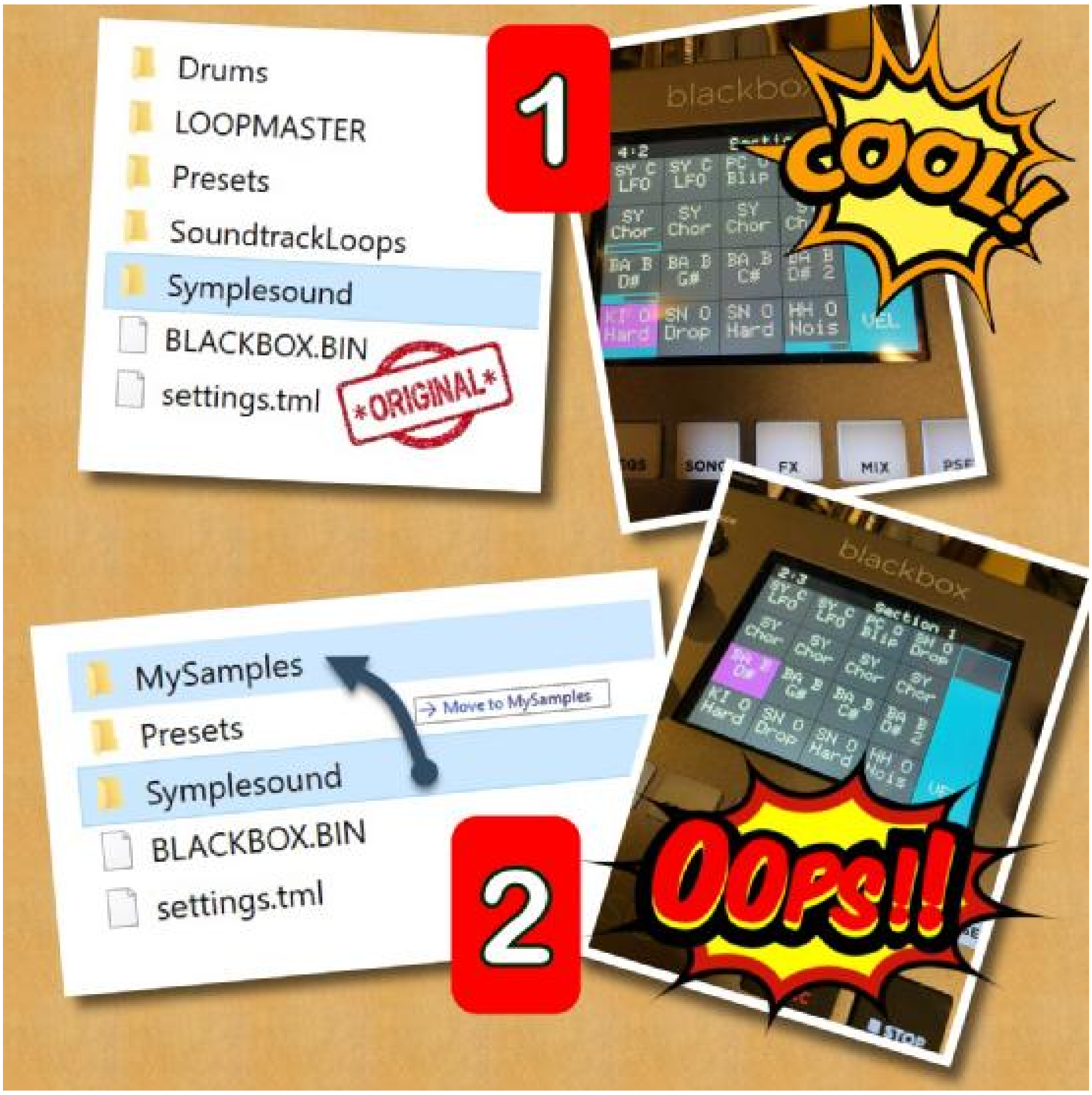
The default folder organization of the Blackbox looks like this:



Name	Size	Type	Modified	Attr
Drums		File Folder	Today 15:54	-------
LOOPMASTER		File Folder	Today 15:49	-------
Presets		File Folder	Today 15:35	-------
SoundtrackLoops		File Folder	Today 15:49	-------
Symplesound		File Folder	Today 15:51	-------
BLACKBOX.BIN	650 KB	BIN File	28.12.2021 16:41	-a-----
settings.tml	690 bytes	TML File	16.02.2022 10:39	-a-----

I have highlighted the samples-folders above. I bet many Blackbox-users quickly have faced challenges with organization of their own sample packs from sound-vendors such as Loopmasters, ADSR Sounds and many more.

Next you will see an illustration where the topmost row (marked by 1), shows the default folder organization of the Blackbox. Note that the Blackbox plays the samples without problem **J**



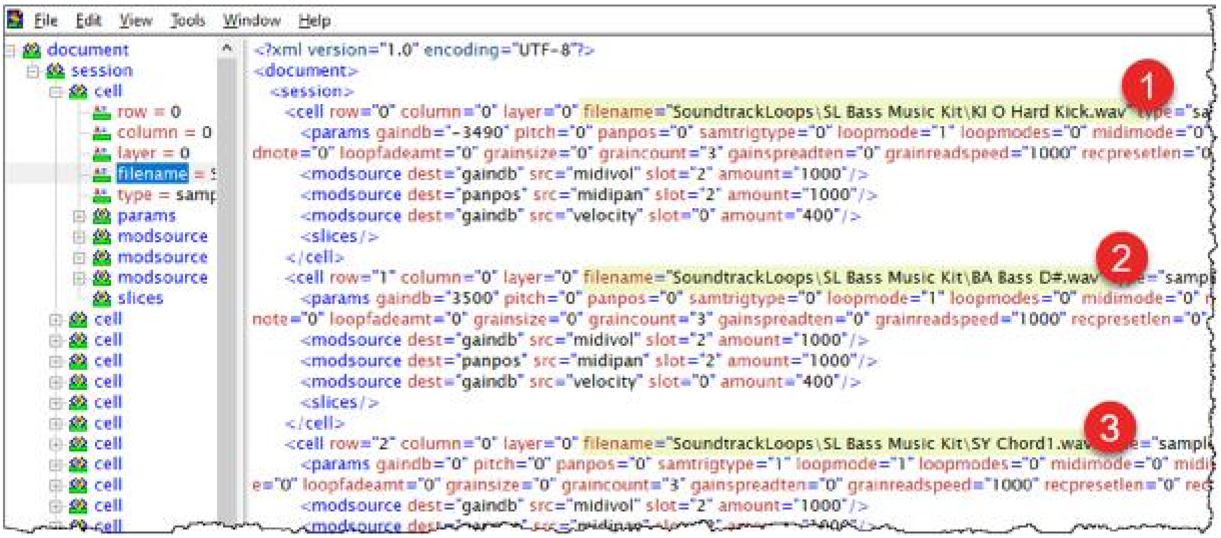
The row at the bottom (marked by 2) in the illustration above, illustrate what happens if you try to move you sample directories into the MySamples-directory. When you load the preset on your Blackbox, the names will be shown as normal, but when you hit the pad, nothing happens. If you have sequences or songs, they will also be silent.

So, if you manually move your content around on the microSD card, you will quickly find that the Blackbox *preset.xml*-files suddenly loose its knowledge about how to access the samples. The reason for this is that the *preset.xml* file contains *references* to the whereabouts of the sample-files themselves.

For example, if you take a look at the default preset found in ...

Presets\000 SL BASS MUSIC DEMO\preset.xml,

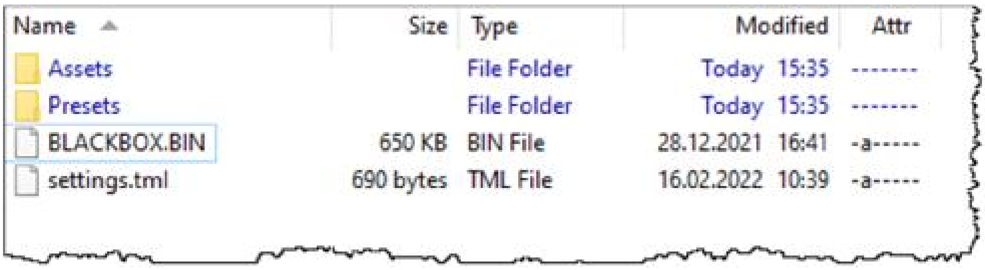
...and open the preset.xml file in a text editor (I use the free and superb firstObject XML Editor from firstObject. Old but good, but first and foremost, extremely fast. It can load hundreds of MBs without blinking!). Below you see the top part of the preset.xml:



```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <session>
    <cell row="0" column="0" layer="0" filename="SoundtrackLoops\SL Bass Music Kit\KI O Hard Kick.wav" type="sa
      <params gaindb="-3490" pitch="0" panpos="0" samtrigtype="0" loopmode="1" loopmodes="0" midimode="0"
dnote="0" loopfadeamt="0" grainsize="0" graincount="3" gainspreadten="0" grainreadspeed="1000" recpresetlen="0
      <modsource dest="gaindb" src="midivol" slot="2" amount="1000"/>
      <modsource dest="panpos" src="midipan" slot="2" amount="1000"/>
      <modsource dest="gaindb" src="velocity" slot="0" amount="400"/>
      <slices/>
    </cell>
    <cell row="1" column="0" layer="0" filename="SoundtrackLoops\SL Bass Music Kit\BA Bass D#.wav" ="samp
      <params gaindb="3500" pitch="0" panpos="0" samtrigtype="0" loopmode="1" loopmodes="0" midimode="0" r
note="0" loopfadeamt="0" grainsize="0" graincount="3" gainspreadten="0" grainreadspeed="1000" recpresetlen="0"
      <modsource dest="gaindb" src="midivol" slot="2" amount="1000"/>
      <modsource dest="panpos" src="midipan" slot="2" amount="1000"/>
      <modsource dest="gaindb" src="velocity" slot="0" amount="400"/>
      <slices/>
    </cell>
    <cell row="2" column="0" layer="0" filename="SoundtrackLoops\SL Bass Music Kit\SY Chord1.wa ="sampl
      <params gaindb="0" pitch="0" panpos="0" samtrigtype="1" loopmode="1" loopmodes="0" midimode="0" midi
e="0" loopfadeamt="0" grainsize="0" graincount="3" gainspreadten="0" grainreadspeed="1000" recpresetlen="0" rec
      <modsource dest="gaindb" src="midivol" slot="2" amount="1000"/>
```

I have highlighted and numbered the three topmost files, and you see the XML element *filename* refers to a file by its full path and file name. This means that for example the file “KI O Hard Kick.wav” marked by 1 above must exist in the folder “SoundtrackLoops\SL Bass Music Kit”. If the Blackbox does not find the file there, the pad will not play!

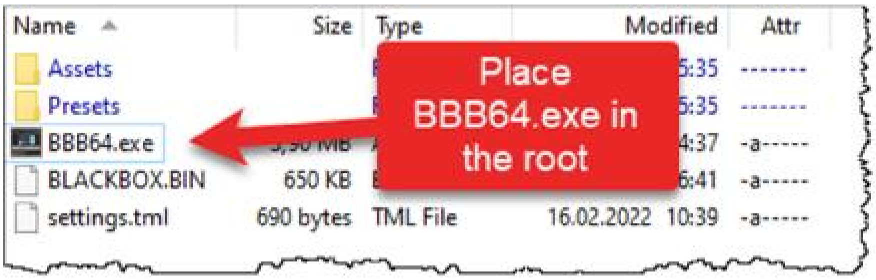
I want to be able to organize the content of my microSD-card the way I want. For example, I want to have all my samples in a separate *Assets*-folder while the mandatory *Presets*-folder stays where it must be. Below you see a screenshot of the root of my Blackbox microSD card:



Name	Size	Type	Modified	Attr
Assets		File Folder	Today 15:35	-------
Presets		File Folder	Today 15:35	-------
BLACKBOX.BIN	650 KB	BIN File	28.12.2021 16:41	-a-----
settings.tml	690 bytes	TML File	16.02.2022 10:39	-a-----

Nice and tidy.

By placing the Blackbox Buddy executable (BBB64.exe) in the root of the microSD card, like this:



Name	Size	Type	Modified	Attr
Assets		F	5:35	-------
Presets		F	5:35	-------
BBB64.exe	5,90 MB	A	4:37	-a-----
BLACKBOX.BIN	650 KB	E	6:41	-a-----
settings.tml	690 bytes	TML File	16.02.2022 10:39	-a-----

… now, simply run it with the command:

```
bbb64
```

BBB will fire up and it will scan all your directories and look for samples and preset.xml files. You will get a report back what needs to be fixed (yes, it runs in a non-destructive *analysis-mode* by default). The report you see scrolling on the screen, is automatically saved to a log for you see with a text editor later. More on that soon (Hint: A really quick way to see your last log is to use `bbb64 –seelog`).

If you are ready write the potential changes to the preset.xml-files, you can run it with the command:

```
bbb64 -write
```

If a preset.xml-file is changed, a backup of the original is created automatically. A bunch of other command line arguments exists, and get a glimpse of those, use the command:

```
bbb64 -help
```

The Blackbox Buddy was first and foremost created to allow me to move my assets (sound packs, wav files, MIDI files etc) around, and keep my Blackbox preset.xml-files automatically updated!

This means that I can easily rename folders without risking that the preset will not play in the Blackbox due to files being moved around.

Secondly, the Blackbox Buddy will also look for any Ableton Live Set preset.als-files. The preset.als will be fixed so they can be opened again in Ableton Live, and they will have their file-references updated in the same manner as for the Blackbox-presets.

Blackbox Buddy makes it *more possible* for me to follow concepts like the ones presented in "The Producer's Guide to File & Sample Organization" 😉

Thirdly, Blackbox Buddy can report many details, for example on missing files. This can make you aware of inconsistencies in your presets and files long before your stand on a scene and do not understand why a preset would not load, or a pad would not play.

That is why!

Introduction and some history

I *love* the 1010music Blackbox sampler!

Super powerful, very versatile and easy to use. I will not dive into all the pros and cons of this sampler but do search the net for reviews. It is worth mentioning that 1010music hosts a nice Blackbox-forum. They are normally very attentive to questions from users, and many very knowledgeable users frequent there too. Highly recommended!

The Blackbox uses a microSD card to hold all the samples and presets. If it is of decent quality (see the forum by the way for a list of recommended cards!), the Blackbox can use it.

One small con with Blackbox is in my view that the Blackbox cannot connect directly to a computer to work with the content of the microSD card. Its USB connector is only for connected music-related USB controllers such as Novation's launchpads or keyboards. The main purpose is that you can use external controllers to control the Blackbox. This is super-cool, but …

You have to extract the microSD card from the Blackbox, and connect it to your computer via some sort of USD-card reader in order to access the content from your computer. Do any modifications and reinsert it into the Blackbox.

The Blackbox has very few restrictions on how you want to organize the content of your microSD card, but the so-called *Presets* must be in the *Presets*-folder on the root of the card.

Inside the Presets-folder each Preset have their own folder with the *preset.*xml, along with optional sample files (which may reside in even deeper folders within the preset-folder itself).

You may also see a *preset.als*-file, which is an Ableton Live Set file that Blackbox have created for you, if you choose to *pack* your preset. Unfortunately, the Blackbox adds a terminating zero to the als-file, so it will not open properly in Ableton Live unless you manually remove the terminating zero yourself. It is easy however, since the preset.als file is an XML file. This means that you can open it and change the content with an editor like Notepad.

If you want your card to be able to upgrade the firmware of the Blackbox, the file BLACKBOX.BIN also must reside in the root of the card.

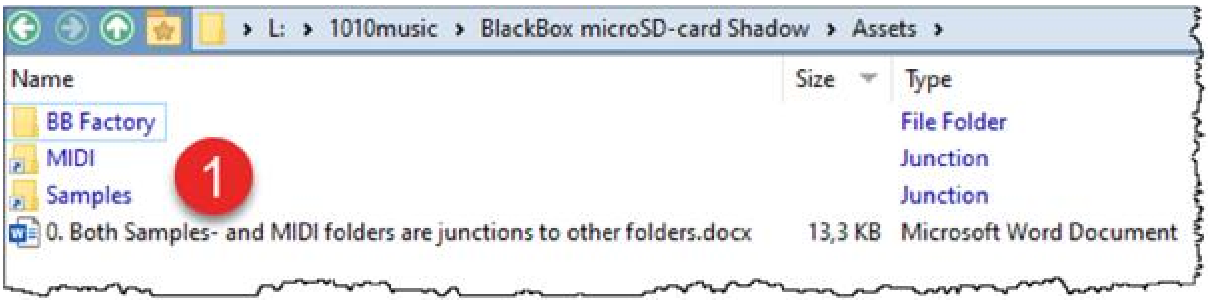
Using a “shadow folder” for the microSD card

Of course, you can work directly on the microSD card. However, I like to have a synchronized “shadow folder” of the microSD card, so I can access the very same assets and Ableton Live set-files as the Blackbox itself. Besides, it acts as a one-level backup too.

By using a synchronization tool like for example Allway Sync, it is a breeze to keep the shadow folder and the microSD card synchronized.

Another little tip worth mentioning is that I use so-called *Windows Junctions*, to create a *reference* to my real sample folder on my hard disks. As we all know, a sample pack folder can quickly grow in both size and number of files, and I like to have *one master folder* for all my samples. The Blackbox shadow-folder simply contain junction to this master folder.

Below you see how this looks like in my Assets-folder of the shadow-directory:



Name	Size	Type
BB Factory		File Folder
MIDI		Junction
Samples		Junction
0. Both Samples- and MIDI folders are junctions to other folders.docx	13,3 KB	Microsoft Word Document

Marked by 1 above, you see that both the *MIDI* and *Samples* are junctions to the respective master folders located outside the current disk.

I used this page for Windows Junctions reference .

Very briefly, the process is as follows In order to create a junction from my source directory “M:\Loops and Samples” to my target directory “Samples” in “L:\1010music\BlackBox SD-card Shadow\Assets”, I used this command in an elevated command prompt (with Administrator rights):

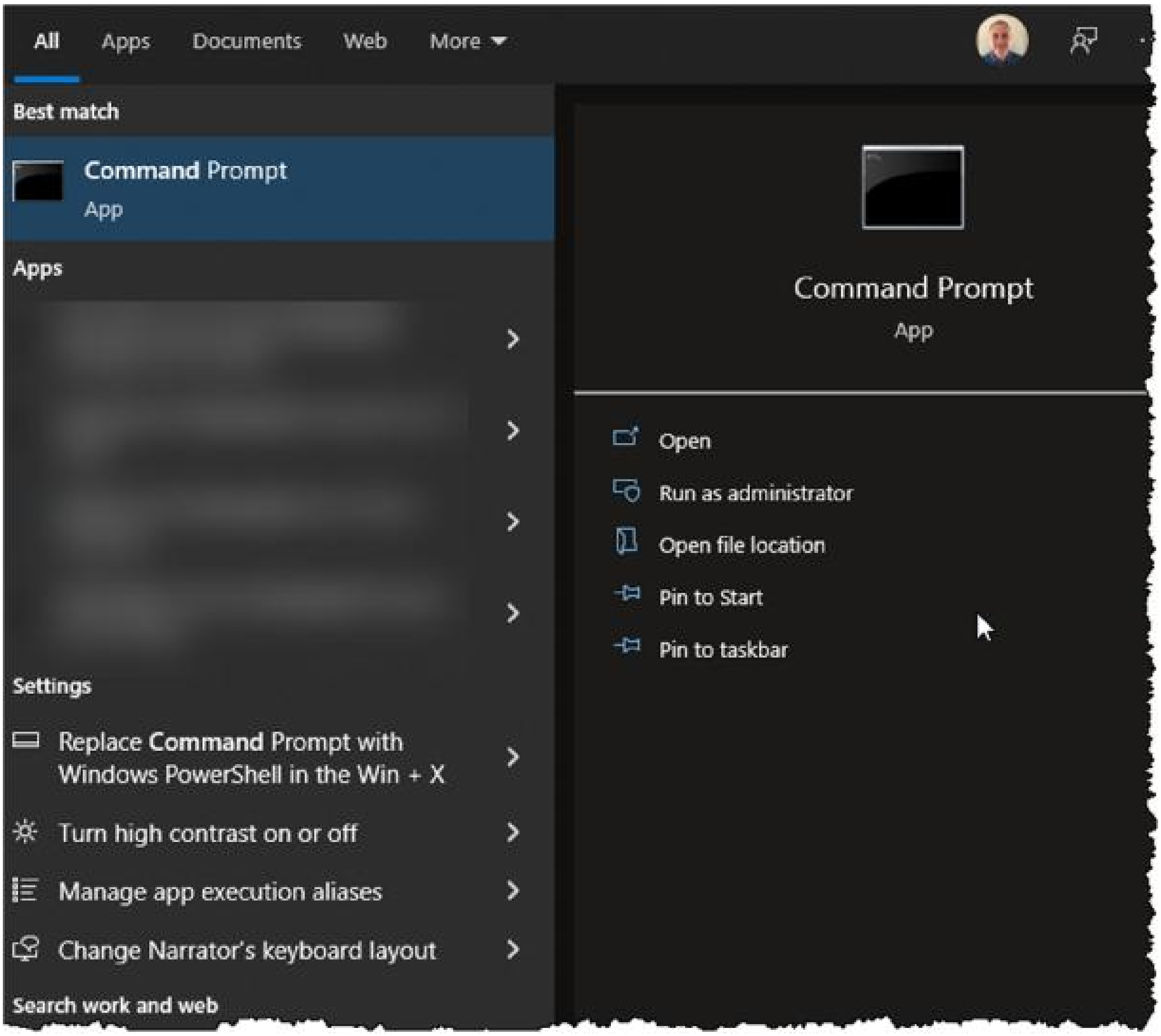
```
mklink /J "M:\Loops and Samples" "L:\1010music\BlackBox SD-card
Shadow\Assets\Samples"
```

If I ever need to delete the junction again (removing the symbolic connection to the source directory), I can simply delete it with the following command in an elevated command prompt:

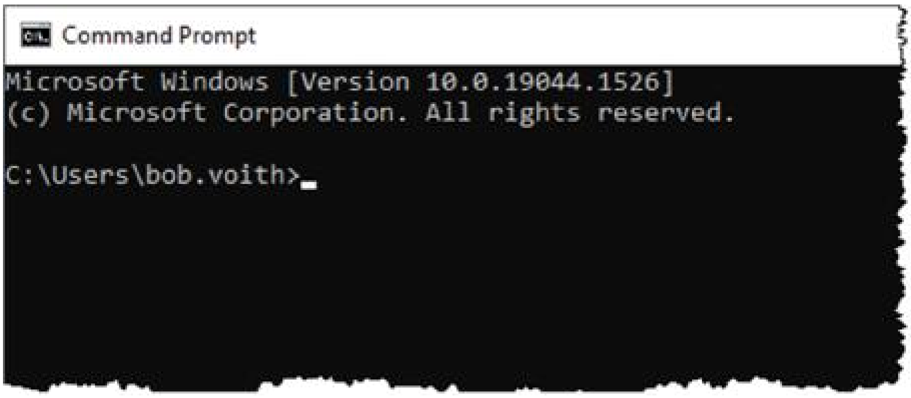
```
rmdir Samples
```

How to use Blackbox Buddy

The BBB64.exe is a Windows-only, console application, which means that you have to open a Command Prompt and issue the commands from the root of the microSD (or the root of the shadow folder, if you use that as described above). Hit the Windows Start-button, and start typing *command* . This will bring up the Command Prompt like this in Windows 10:



Click Open and a Command Prompt will open like this:



```
Microsoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bob.voith>_
```

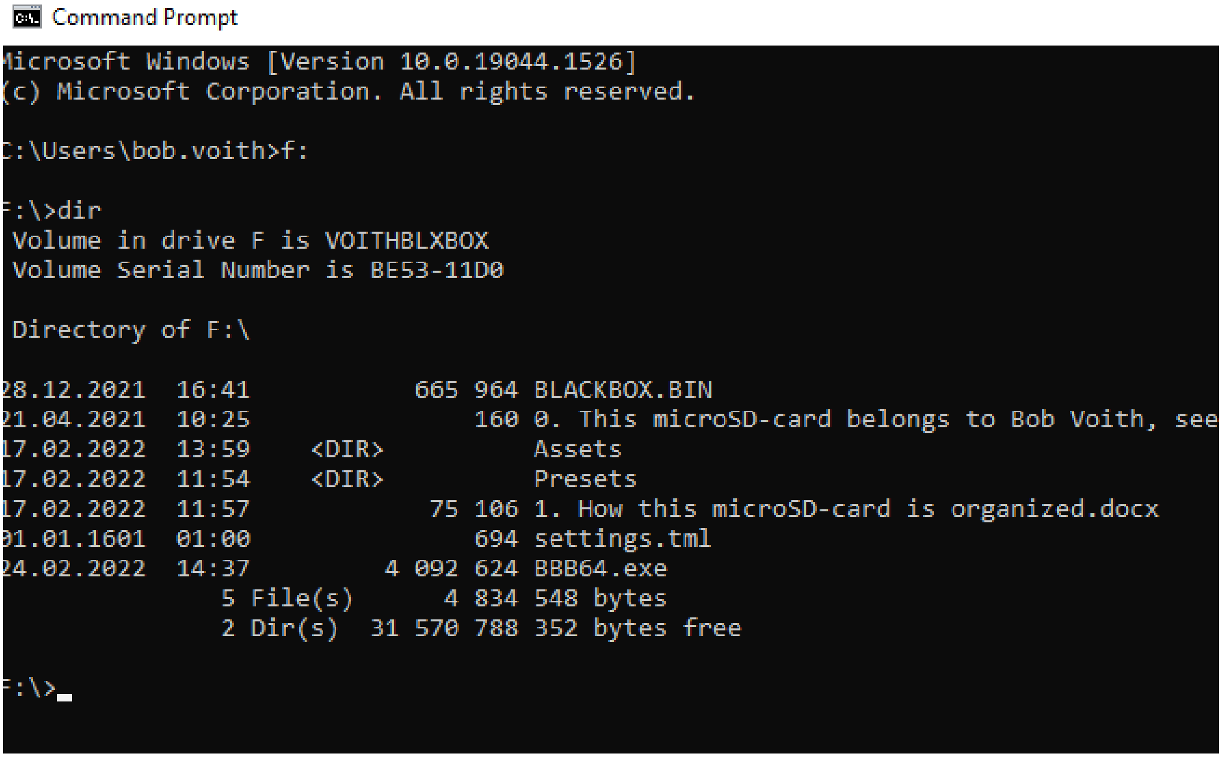
By default, your current folder will be your user folder. Use the old and trusty DOS commands to navigate to your microSD-card. On my machine, the microSD card has been mounted as F: and I go to the F-drive simply by issuing the following command (ensure you also have the last colon before pressing enter):

```
F:
```

You can verify that you actually are in the correct place, by using the following command:

```
dir
```

This will show you the content of the F-drive, like this (note that you see *my content*. Yours may of course differ):



```
licrosoft Windows [Version 10.0.19044.1526]
(c) Microsoft Corporation. All rights reserved.

:\Users\bob.voith>f:

:\>dir
 Volume in drive F is VOITHBLXBOX
 Volume Serial Number is BE53-11D0

 Directory of F:\

28.12.2021  16:41           665 964 BLACKBOX.BIN
21.04.2021  10:25               160 0. This microSD-card belongs to Bob Voith, see
17.02.2022  13:59    <DIR>          Assets
17.02.2022  11:54    <DIR>          Presets
17.02.2022  11:57            75 106 1. How this microSD-card is organized.docx
01.01.1601  01:00               694 settings.tml
24.02.2022  14:37         4 092 624 BBB64.exe
               5 File(s)      4 834 548 bytes
               2 Dir(s)  31 570 788 352 bytes free

:\>_
```

Note that you also should see the BBB64.exe placed in the root, as shown above.

Now you are ready to do some mojo on your Blackbox microSD-card!

By default, BBB64 will run in *analysis mode* meaning that it run as normal, but do not write anything to disk. Thus, you cannot destroy anything simply by running the application.

The most important command line parameter, besides just running it, is the following:

```
bbb64 -help
```

It will show you a very brief help listing the different command line parameters. If you get irritated that the top of the help message scrolls off the screen, you can use the following command:

```
bbb64 -help | more
```

This will show you the first page and you have to click a key to continue to the next page. Click CTRL + C to abort the paging.

You can also save the help to a file, by redirecting the help message to a text file with the following command:

```
bbb64 -help >bbb.txt
```

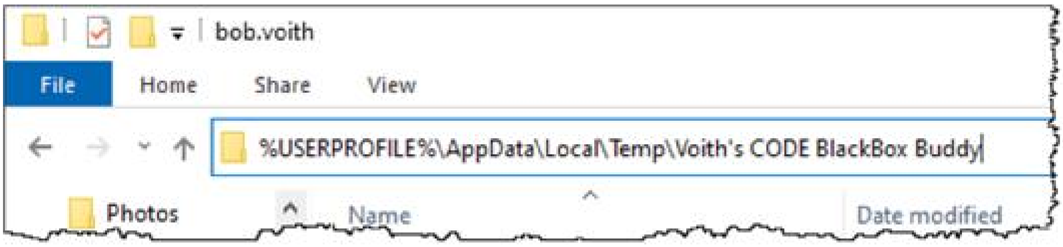
You will then find a text file with the name bbb.txt

The process

The BBB64 do separate things in sequence in order to do its job. By default, BBB will also create a log file with the same content as you see on the screen. The default placement of the log file is in your temporary directory. In *Windows Explorer*, you can access this by using the following path:

```
%USERPROFILE%\AppData\Local\Temp\Voith's CODE BlackBox Buddy
```

...like this:


bob.voith
File
Home
Share
View
%USERPROFILE%\AppData\Local\Temp\Voith's CODE BlackBox Buddy
Photos
Name
Date modified

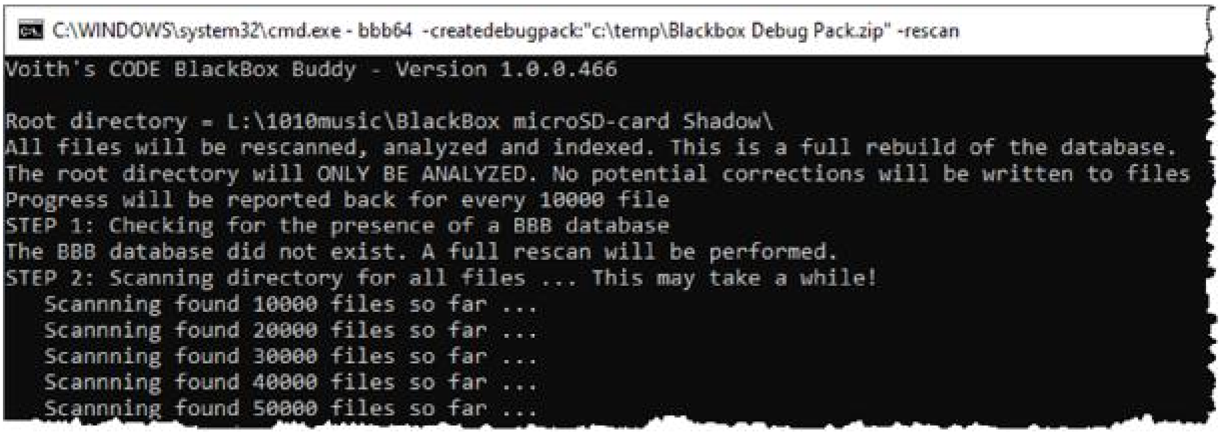
Inside the temporary folder, you will find all the logs and the Blackbox Buddy-database itself. The BBB-database is an ordinary XML file so you can peek at it with a text-editor or a XML editor. The logs do have a date and time prefix to help you separate the runs.

A really quick way to see the latest log, is to use `bbb64 -seelog`

If you forget about the parameters, then use `bbb64 –help` to show both the help, and the whereabouts of your directories.

The steps

Below you see how the command prompt looks when BBB have started:


C:\WINDOWS\system32\cmd.exe - bbb64 -createdebugpack:"c:\temp\Blackbox Debug Pack.zip" -rescan

```
Voith's CODE BlackBox Buddy - Version 1.0.0.466

Root directory = L:\1010music\BlackBox microSD-card Shadow\
All files will be rescanned, analyzed and indexed. This is a full rebuild of the database.
The root directory will ONLY BE ANALYZED. No potential corrections will be written to files
Progress will be reported back for every 10000 file
STEP 1: Checking for the presence of a BBB database
The BBB database did not exist. A full rescan will be performed.
STEP 2: Scanning directory for all files ... This may take a while!
    Scannning found 10000 files so far ...
    Scannning found 20000 files so far ...
    Scannning found 30000 files so far ...
    Scannning found 40000 files so far ...
    Scannning found 50000 files so far ...
```

1. Step 1 will look for an existing BBB-database. The first time you run BBB, it will of course not find this database, and BBB will automatically perform the following steps for you

2. Step 2 will scan all directories and files, starting from the position of the BBB64.exe.

   If you have started BBB for the first time, or you use `bbb64 –rescan`, BBB will scan all directories and files starting from the root directory. If you have followed along so far, you have placed the BBB64.exe in the root of the microSD-card. This means that BBB scans all content of the card.

   For subsequent runs, BBB will only scan the *Presets*-folder. This means that BBB will combine preprocessed whereabouts of all you assets, and assume that you do most of your preset-changes inside the Presets-folder.

   If you keep your Blackbox- and Ableton presets in other directories than *Presets*, you need to run BBB with the –rescan parameter in order to process presets in other directories.

3. Step 3 will analyze the files from the previous steps. As for the steps above, this takes a while the first time BBB runs. In subsequent runs, BBB only needs to analyze the content of the Preset-folder, which normally is much faster.

   The analysis involves analyzing the directory- and file name structure, preparing for the upcoming Index-step

   The BBB database is stored in the your temporary directory.

4. Step 4 indexes all the stuff from the previous steps, no matter if it is the first time you run BBB, using `–rescan` or a subsequent run.

   This involves creating an index for extremely fast lookup during the upcoming check-and-potentially-fix phase for the Blackbox and Ableton preset files.

   While using an index is very fast when created, it uses computer memory to create the index. This can be quite big if you have many assets in your directory. You might therefore experience memory problems if you run the BBB on a machine with little memory.

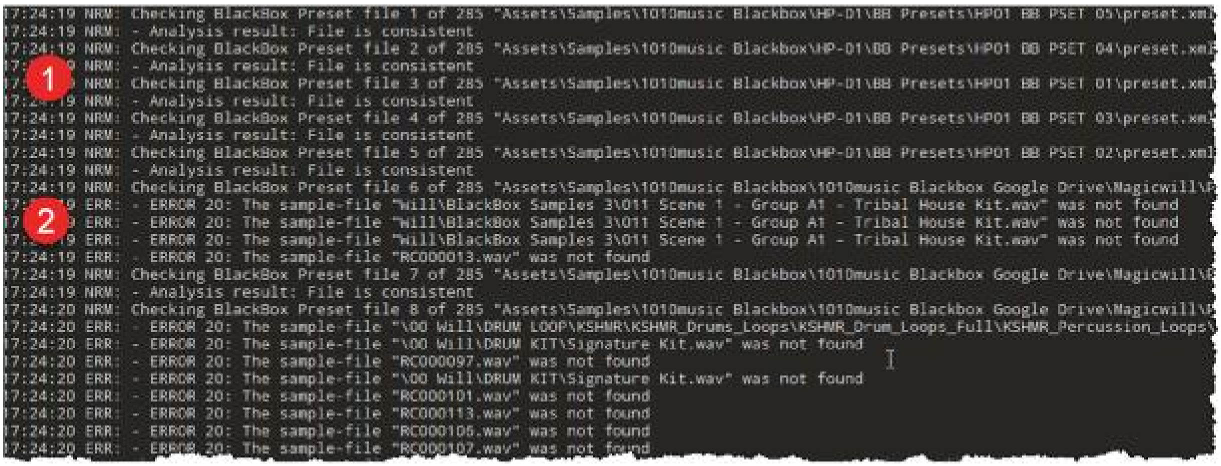
   To give you an idea of how fast BBB is, these are the times on my machine right now, for a test card with more than 700.000 files

   Initial run – or –rescan:
   Step 1: Loading database: 0 Seconds, as nothing needs to be loaded
   Step 2: Scanning: Approximately. 2 minutes
   Step 3: Analyzing: Approximately 2 minutes
   Step 4: Indexing: Approximately 25 seconds

Subsequent runs:
Step 1: Loading database: 1 Second (pretty good for 245 MB database!)
Step 2: Scanning: Approximately. 2 seconds
Step 3: Analyzing: Approximately 15 seconds
Step 4: Indexing: Approximately 20 seconds

Your mileage may of course differ from this, because your computers configuration (amount of RAM, SSD or HDD, processor type and whatnot) probably differs from my computer. However, the stats above should give you a hint of the differences between first- and subsequent runs and that you can have a pretty decent BBB-performance even if you have many samples and presets.

5. The subsequent steps involves processing presets. By default, BBB starts first with Blackbox-presets. The reports from the Blackbox-processing will normally generate several lines such as for example:



```
17:24:19 NRM: Checking BlackBox Preset file 1 of 285 "Assets\Samples\1010music Blackbox\HP-01\BB Presets\HP01 BB PSET 05\preset.xml
17:24:19 NRM: - Analysis result: File is consistent
17:24:19 NRM: Checking BlackBox Preset file 2 of 285 "Assets\Samples\1010music Blackbox\HP-01\BB Presets\HP01 BB PSET 04\preset.xml
17:24:19 NRM: - Analysis result: File is consistent
17:24:19 NRM: Checking BlackBox Preset file 3 of 285 "Assets\Samples\1010music Blackbox\HP-01\BB Presets\HP01 BB PSET 01\preset.xml
17:24:19 NRM: - Analysis result: File is consistent
17:24:19 NRM: Checking BlackBox Preset file 4 of 285 "Assets\Samples\1010music Blackbox\HP-01\BB Presets\HP01 BB PSET 03\preset.xml
17:24:19 NRM: - Analysis result: File is consistent
17:24:19 NRM: Checking BlackBox Preset file 5 of 285 "Assets\Samples\1010music Blackbox\HP-01\BB Presets\HP01 BB PSET 02\preset.xml
17:24:19 NRM: - Analysis result: File is consistent
17:24:19 NRM: Checking BlackBox Preset file 6 of 285 "Assets\Samples\1010music Blackbox\1010music Blackbox Google Drive\Magicwill\F
17:24:19 ERR: - ERROR 20: The sample-file "Will\BlackBox Samples 3\011 Scene 1 - Group A1 - Tribal House Kit.wav" was not found
17:24:19 ERR: - ERROR 20: The sample-file "Will\BlackBox Samples 3\011 Scene 1 - Group A1 - Tribal House Kit.wav" was not found
17:24:19 ERR: - ERROR 20: The sample-file "Will\BlackBox Samples 3\011 Scene 1 - Group A1 - Tribal House Kit.wav" was not found
17:24:19 ERR: - ERROR 20: The sample-file "RC000013.wav" was not found
17:24:19 NRM: Checking BlackBox Preset file 7 of 285 "Assets\Samples\1010music Blackbox\1010music Blackbox Google Drive\Magicwill\F
17:24:19 NRM: - Analysis result: File is consistent
17:24:20 NRM: Checking BlackBox Preset file 8 of 285 "Assets\Samples\1010music Blackbox\1010music Blackbox Google Drive\Magicwill\S
17:24:20 ERR: - ERROR 20: The sample-file "\00 Will\DRUM LOOP\KSHMR\KSHMR_Drums_Loops\KSHMR_Drum_Loops_Full\KSHMR_Percussion_Loops\
17:24:20 ERR: - ERROR 20: The sample-file "\00 Will\DRUM KIT\Signature Kit.wav" was not found
17:24:20 ERR: - ERROR 20: The sample-file "RC000097.wav" was not found
17:24:20 ERR: - ERROR 20: The sample-file "\00 Will\DRUM KIT\Signature Kit.wav" was not found
17:24:20 ERR: - ERROR 20: The sample-file "RC000101.wav" was not found
17:24:20 ERR: - ERROR 20: The sample-file "RC000113.wav" was not found
17:24:20 ERR: - ERROR 20: The sample-file "RC000106.wav" was not found
17:24:20 ERR: - ERROR 20: The sample-file "RC000107.wav" was not found
```

Above you see a bunch of lines marked by 1, which all seems to be OK.

You get the message “Analysis result: File is consistent”. This means that all the referenced files within the Blackbox preset.xml-file has been found.

Marked by 2 above, you see how BBB reports errors. For example, the “ERROR 20: The sample-file "RC000013.wav" was not found” means that the referenced sample file RC000013.wav was not found.

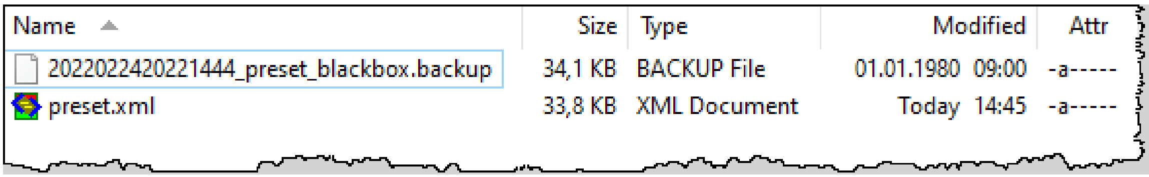
Remember that BBB normally process the files *very fast* after the initial first three steps. This means that you probably need to open the log when BBB has finished (Hint: bbb64 –seelog)

Backups

BBB will indeed change the content of your files if you use the command line parameter `-write`. To be sure that you can revert back to the previous version of the file, BBB will by default, create a backup of the original file before it writes its changes to disk. For example, in the Preset-directory …

“Presets\000 SL BASS MUSIC DEMO”

… you may see the following content after BBB has run (and created a backup):



Name	Size	Type	Modified	Attr
2022022420221444_preset_blackbox.backup	34,1 KB	BACKUP File	01.01.1980 09:00	-a-----
preset.xml	33,8 KB	XML Document	Today 14:45	-a-----

Note that BBB always store the backup in the same directory as the file it changes. For example can a backup be shown in a folder like “Presets\000 SL BASS MUSIC DEMO” like this:

The preset.xml is the current and corrected Blackbox preset-file, while the `20220224202214_preset_blackbox.backup` is the previous version of the preset.xml.

You can simply revert to the previous file by deleting the preset.xml, and rename the backup file to `preset.xml`. This is easy enough if you want to revert for a single preset file. However, what if BBB changed hundreds of presets, and you want to revert all of them? A couple of backup-related command line parameters comes to rescue, namely the `-listbackup` and `-revertbackup:<timestamp>`. By using these parameters, you can quickly revert back to any earlier level of preset.xml for any files changed during that BBB-session.

The Command Line Parameters

You have already seen command line parameters `like -help, -rootdir` and `-write` mentioned before. In this section, you will see all available command line parameters.

Please note that some parameters act as *flags*, meaning that you only have to specify the parameter in order to trigger something. The `-help` parameter is an example of a flag.

Other parameters expect an extra, eh… parameter. For example, the `-rootdir` must have an extra parameter specifying *which directory* to use. When a parameter needs a parameter, you separate them with a colon. Below you see an example of the `-rootdir` with a parameter:

```
bbb64 -rootdir:”c:\temp\My BlackBox Content”
```

Note how the colon separate the parameter and its parameter.

Finally, some parameters can be both a flag and a parameter with a parameter. This will be shown below by examples.

Parameter	Description
-analyze	By default, BBB will run in analysis mode, scanning and checking everything, but not saving anything to disk. The default behavior is therefore completely non-destructive. This flag is seldom used, but it forces the analysis mode set. Note that if you also specify the -write parameter, the -analyze parameter win!
-rootdir:<path>	Set the root directory to another directory that the current position of BBB64.exe.

	By default, BBB will regard the position where itself (BBB64.exe) is placed, as the root directory. By specifying a <path> like for example: `-rootdir:”c:\Temp\My Blackbox Content”` BBB will now use the specified directory instead. Note how the apostrophes enclose the path since it contains spaces.

-logfile:<path and file name>	By default, BBB will create a log file in your temporary directory. You can access this directory with the following in Windows Explorer:
	`%USERPROFILE%\AppData\Local\Temp\Voith's CODE BlackBox Buddy`
	By specifying -logfle, you can output the log to whatever path and file name you want. For example:
	`-logfile:”c:\temp\Blackbox Buddy.log”`
	Note how the apostrophes enclose the path and file name, since it contains spaces.
	Tips: Use `bbb64 –help` to see the directories on your machine!

-loglevel:<level>	BBB has different levels of messages, starting with Errors. Then comes Warnings, Normal messages, Verbose messages and finally Debug-messages. Errors will always be shown no matter what setting you use.
	The levels are (the abbreviation seen in the logs are shown in parenthesis):
	`0 : Errors (ERR)`
	`1 : Warnings (WRN)`
	`2 : Normal (NRM, the default)`
	`3 : Verbose (VBR)`
	`4 : Debug (DBG)`
	
	For example:
	`-loglevel:3`
	
	By default, BBB uses level 2, normal.

-nolog	You don’t want any log files at all? By using this flag, you can turn off writing to log files completely. The console will be updated as normal and pay attention to the `–loglevel`-setting too.

-reportprogress:<number>	How often should BBB report that it actually is working?
	By default, BBB will report for every 10.000 file it is processing. You will see messages on screen like:
	`Analyzing file #10000 of 240000`
	`Analyzing file #20000 of 240000`
	`Analyzing file #30000 of 240000`
	… etc.
	By using this parameter, you can change the interval BBB reports back. For example, the parameter below ensures that BBB reports back for every 15th file:
	`-reportprogress:15`

-version	This flag shows the version of BBB. No further processing will be done

-help	Show the brief help. To page the output, you can use the pipe command like:
	`bbb64 -help | more`
	To dump the content to a file, use redirect like:
	`bbb64 -help > filename.txt`

-nobackup	By default, BBB will create a backup of any files it changes. See the earlier section about Backup for more info. This flag turn off the generation of backup files! Yes, this is only for the strong-hearted people out there. One reason for doing this might be if you are tight on free storage space on your microSD-card.

-listbackup	List all the backup timestamps that exists in the root directory and all sub folders. Its purpose is to show you which time stamps you can use for the next parameter, the `-revertbackup`.
	For example:
	`-listbackup`
	… may result in the following output:
	`20220224180117`
	`20220224202214`
	You now know that you have two BBB-runs that have generated backup files. The timestamps above can be used with the next parameter, `-revertbackup`.

-revertbackup:<timestamp>	Revert an earlier version of the preset file, back to the current version of preset.xml and preset.als. The parameter ensures that all backup files throughout the directories, with the same time stamp, are reverted.
	
	Use -listbackup to list backup time stamps first. For example:
	`20220224180117`
	`20220224202214`
	
	You choose the timestamp you want to use, for example the latter:
	`-revertbackup:20220224202214`
	All backup files with that timestamp, will now become preset.xml or preset.als. The backup files is then removed automatically.

-clearbackups	This flag will search for all backup files (you can however limit the search with –`skipbb` or –`skipals`). Any found backup files will be deleted! Remember that this will remove any possibility of reverting back to previous backup if needed. This should this therefore only be used when you are complete sure that everything is OK.
-listfiles:<DOS wild card>	Looking for a file or a set of files in your directory structure? By using a DOS wild card, like `*.*` for all files, BBB will list them for you. Another example: `-listfiles:*.wav` to see all wav files

	`-listfiles:”Kick 2*.*”` – Any file starting with “Kick 2”
-processpattern:<pattern>	By using a text pattern, you can limit what presets you work with. For example: `-processpattern:”8BIT”` … will process only presets that have the string “8BIT” in their names. Note that this parameter works for both Blackbox- and Ableton Live presets. If you need to work with just Blackbox or Ableton Live sets, then use one of the `-skipbb` or `-skipals`
-skipbb	This flag enable you to limit what to process, and it will skip Blackbox-presets processing during analyze, write, clear backups and revert backup.
-skipals	This flag enable you to limit what to process, and it will skip Ableton-presets processing during analyze, write, clear backups and revert backup.
-createdebugpack:<path and file name of zip file>	Create a zip file containing all the directories and file names from your original directory. Why? Because BBB works with directory structures with all its files. To do this it really only needs the preset-files (preset.xml and preset.als) and the directory- and file name of the samples. If BBB detects that a sample has moved, it will correct the Blackbox preset.xml or Ableton Live preset.als. The main purpose is to experiment with real directory structures and real file names. Note that this parameter will create so-called zero-byte files, meaning that they are completely empty. The only files that is copied with their content, is the preset-files (preset.xml and preset.als) as well as the generated BBB-backup files. *It is very important to understand, that this feature is by no means, a backup feature!* For example: `-createdebugpack:”c:\temp\BlackBox Debug Pack.zip”`
-seelog	This flag will lookup the *latest* log-file created by BBB, and launch it with your default editor for text. It is meant as a quick way of taking a look at the messages from the latest run.
-clearlogs	This flag will search for all BBB-logs in the temporary directory, and delete them.
-register:<license key>	Whoooho! You have gone the extra mile and actually donated an amount to the future development of BBB! Thank you! This parameter can be used to enter the license key you receive in the registration e-mail from vcode.no.

	If you don’t receive an registration e-mail within 24-48 hours, please check your spam-filter. See the chapter “How to enter the license” for more information

How to enter the license

The BBB is completely free to use – as much as you want. Besides a small “Want to donate?”-string, there is absolutely no differences between the never-ending trial version, and the registered version of BBB.

If you want to donate, head over to https://www.vcode.no/bbb and look for the Donate-link. Remember, you decide yourself what amount to donate!

In return, you will receive a registration e-mail with a license key, which can be installed on your computer. It will remove the donation-reminders when you run BBB.

The license key is a long sequence of characters and numbers. The whole string should be used with the parameter `–register`, like shown below:

```
bbb64 –register:<here goes the whole string>
```

A more realistic example is shown below:

```
bbb64 -register:4C95CF70F30A23AA38C…veeeeerrrry…looooong……key46911FB427301BE1
0B2D07A791B68BB9F8276519AF00602F7D255C216DE741B1821D512871BBD2810BDD479E64049
112F9E2E3EEC63207BA5C8C5B14EF9B18679C2AE6BBBF3B10795E9AC577ADF2C95CF70F30A23F
F38CF4671D10259809E5B81F16EF80D976362C5EEA325763C23078B7A291043FD2492BF29D69F
5D159851B26D2EB9357D3DD6A6FF28AF313748111795260F4385581B6AD209392FFCD200C0146
50E9807B3C112A01E25EF5F0F0FA9EFB4E3F3D408ED95237427269E8A79E148ADADBB33B45A53
FDDF7FC9B93ABCA8DCD3B5B0899B45FA1A6F0A3AB20AC705E4A225EFEFC22210E054A529D4A40
3DEB73C47505C31C5B4966574B1046F3BA342DE91B54E984C71792228781BA4AB7517BF581C0D
C062BFCA85F33CEF29B7F5EE9D3D9D1F82B79A89C94EB044C9B832994FD1009BEE1483ECA577D
```

Behind the scenes, BBB stores the license in the Windows Registry.

If you need to run BBB on different computers, this procedure has to be repeated on each computer.