swissbit®

# SBDM – Swissbit Device Manager Tool

## Users Guide

Date: 18 March 2026
Revision: 1.10

# Content


1 Overview ..... 3
1.1 Supported product platforms ..... 3
2 Deployment and Requirements ..... 4
2.1 Windows ..... 4
2.2 Linux ..... 4
3 Usage of graphical user interface ..... 5
3.1 Launching SBDM ..... 5
3.2 System tray ..... 7
3.3 Settings ..... 7
3.4 Device Details ..... 8
3.5 Tools: Firmware Update ..... 11
4 Usage of command line utility ..... 13
4.1 Launching SBDM ..... 13
4.2 LTM mode ..... 14
4.3 Firmware update mode ..... 15
4.4 Return codes ..... 15
5 Firmware Update Linux Live CD manual ..... 16
5.1 Creating a bootable medium ..... 16
5.2 Bootable USB stick ..... 16
5.3 Update host requirements ..... 17
5.4 Usage ..... 17
6 Troubleshooting ..... 18
6.1 Connecting a device indirectly (via bridge chip) ..... 18
6.2 No drives detected on Linux ..... 19
6.3 No SMART data on Windows for IDE / SATA ..... 19
6.4 SD card with write protection feature not detected ..... 19
6.5 Application is not properly rendered and very slow on embedded devices ..... 19
7 License ..... 20
8 SBDM Revision History ..... 20
9 Document Revision History ..... 21

# 1 Overview

SBDM (Swissbit Device Manager) for Windows is a utility program that reads status information from Swissbit products. It uses S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology) and USB vendor specific commands to gather lifetime information.

It allows reading a number of device information and statistics:

- General device information, e.g. ATA model ID, serial number, firmware version
- Spare Blocks statistics (including number at time of production and current)
- Usage statistics (NAND Flash erase cycles)
- Various ECC / CRC counters
- History of all values is recorded per device

## 1.1 Supported product platforms

The software supports Swissbit devices of the following product platforms:

| | |
|---|---|
| CompactFlash: | C-3x (SMART capable only), C-4x, C-5x |
| SSD SATA: | X-1x, X-2x, X-5x, X-6x, X-7x, X-8x |
| CFast: | F-1x, F-2x, F-5x, F-6x, F-8x |
| USB: | U-1x, U-4x, U-5x, U1x |
| SD: | S-2x (support for Linux native SD interface only), S-4x, S-5x, S-6x |
| eMMC: | EM-2x, EM-3x, M1x (all support for Linux native MMC interface only) |
| NVMe: | N-1x, EN-2x, E2x, N-2x, N2x, G-2x, G2x, N-3x, N3x, N4x, N5x, N7x, D1x, A1x, A2x |

# 2 Deployment and Requirements

## 2.1 Windows

The executables can be run on any supported system without prior installation. SBDM for Windows is provided for two different architectures:

- Windows-x86 32-bit versions for legacy systems
- Windows-x64 64-bit versions

Each folder contains the following executable files:

- SBDM.exe graphical user interface
- sbdm-cli.exe command line user interface

The device database holding the history of each device is generated in the working directory:

- SBDM.db device database

There are some requirements and technical limitations for the use of the software:

- Administrator rights are necessary.
- Windows 7 or higher is needed.
- The device must be connected to a native interface (e.g. IDE / SATA, USB, SD or NVMe). Please check chapter 6.1 Connecting a device indirectly (via bridge chip for additionally supported bridge chips.

## 2.2 Linux

The SBDM can be run on any supported system without prior installation. SBDM for Linux is provided for different architectures:

- Linux-i386 x86 32bit architectures
- Linux-x86-64 x86 64bit architectures (amd64)
- Linux-aarch64 aarch64 (armv8 64bit) ARM architectures
- Linux-armhf armhf (armv7 32bit) ARM architectures

Each folder contains the following executable files:

- SBDM.AppImage graphical user interface
- sbdm-cli command line interface

The device database holding the history of each device is generated in the working directory if it does not exist yet:

- SBDM.db device database

There are some requirements and technical limitations for the use of the software:

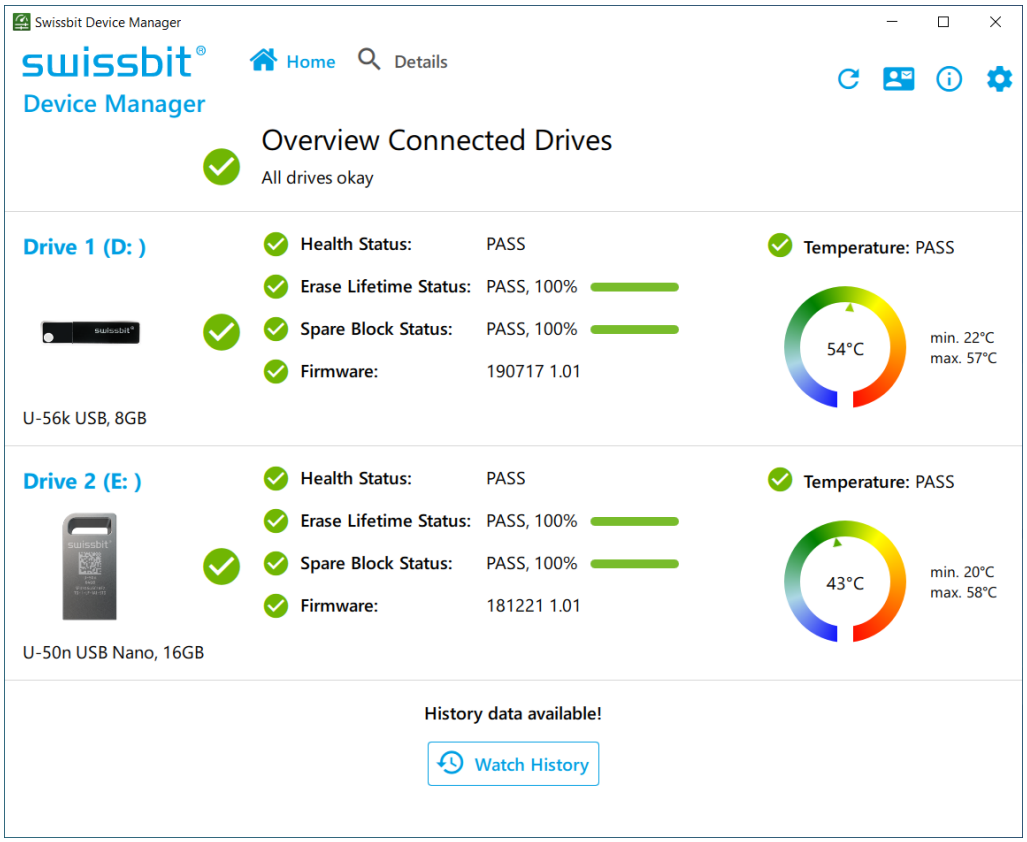
- Root rights are necessary
- Linux Kernel 2.6 or newer is needed
- GUI tool:
  - GLibc 2.3 or newer
  - X11 and Xcb libraries
- The device must be connected to a native interface (e.g. IDE / SATA, USB, SD or NVMe). Please check chapter 6.1 Connecting a device indirectly (via bridge chip for additionally supported bridge chips.

# 3 Usage of graphical user interface

## 3.1 Launching SBDM

### 3.1.1 Windows

To start the application, launch SBDM.exe (execute as administrator). The main screen will display all devices found in the system with the current status:




The drives are listed with the Windows internal drive number.

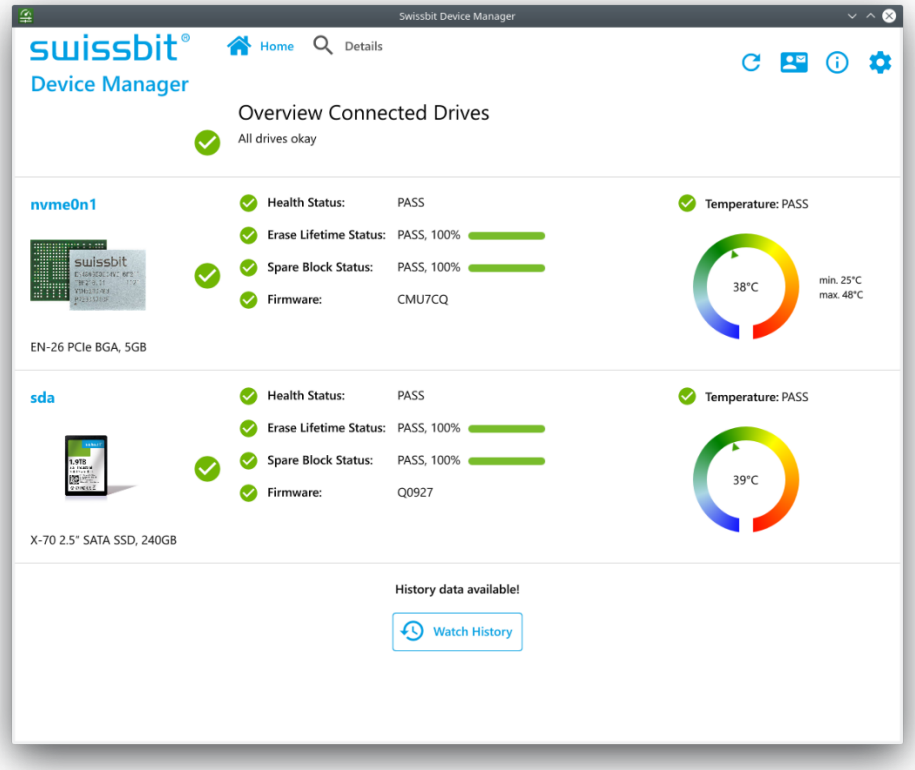
### 3.1.2 Linux

The Linux application is provided as AppImage package that already includes all required libraries. Please visit https://appimage.org/ for more details.

To start the application:

1. Make executable with: "chmod a+x SBDM-<arch>.AppImage"
2. Start as root or with sudo: "sudo ./SBDM-<arch>.Appimage"

The main screen will display all devices found in the system with the current status:



### 3.1.3 Command line arguments

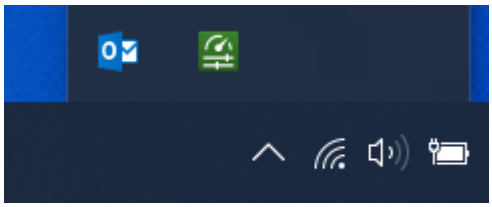
The utility supports the following command line arguments for graphical user interface mode:

```
Usage: ./SBDM [options]
Options:
  -h, --help, -?                Displays help on commandline options.
  -v, --version                 Displays version information.
  -m, --minimized               Utility is started in minimized mode (hidden in
                                system tray)
```

## 3.2 System tray

### 3.2.1 Windows

The SBDM tool can be running in the background, showing in the system tray only.

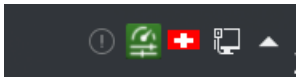


Even when running in the background, it periodically checks the drives in the system and records history data.

To hide the Tool, just use the minimize buttons of Windows, or the "Hide" button. To open the tool up again, left click on the system tray icon.

### 3.2.2 Linux

The SBDM tool can be running in the background, showing in the system notification tray only (where the window manager provides a system tray).

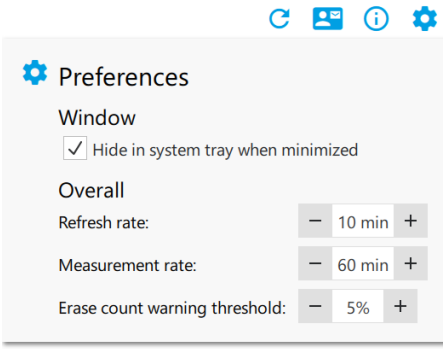


Even when running in the background, it periodically checks the drives in the system and records history data.

To hide the Tool, just use the minimize buttons on the main window, or the "Hide" button. To open the tool up again, left click on the system tray icon.

## 3.3 Settings

The SBDM tool supports changing appearance and monitoring settings.



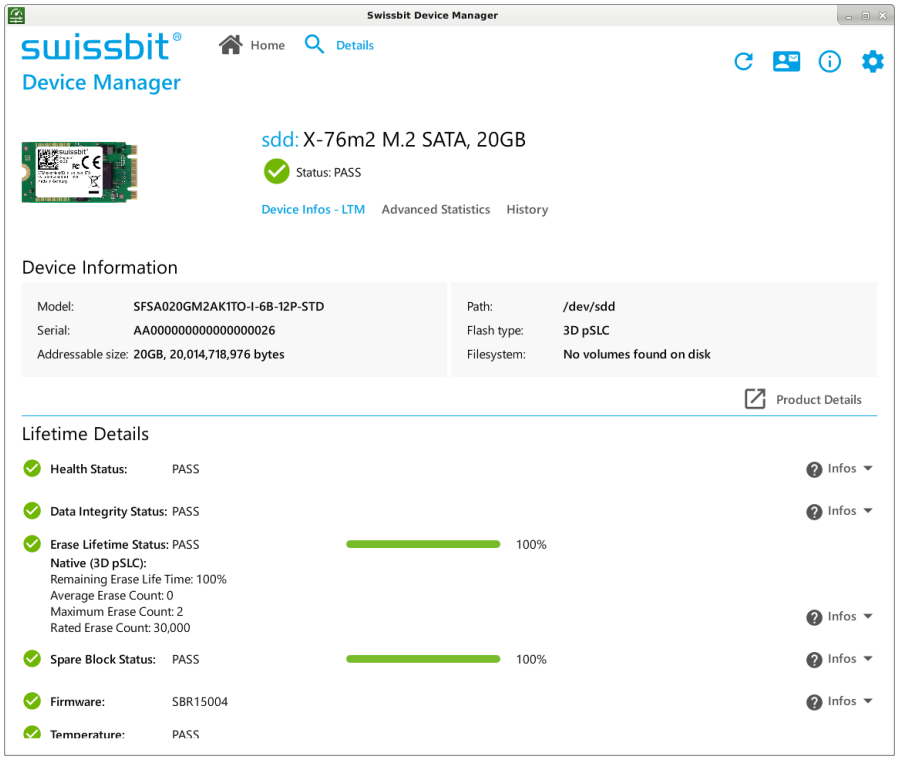

| | |
|---|---|
| Refresh rate: | Sets the periodic time for checking and refreshing life time data. |
| Measurement rate: | Sets the periodic time for storing life time data in the device database file. |
| Erase count warning threshold: | Sets the threshold for the "Erase Lifetime Status" when the SBDM tool will output a warning |

## 3.4 Device Details

The Details page provides all available lifetime information for a selected device.

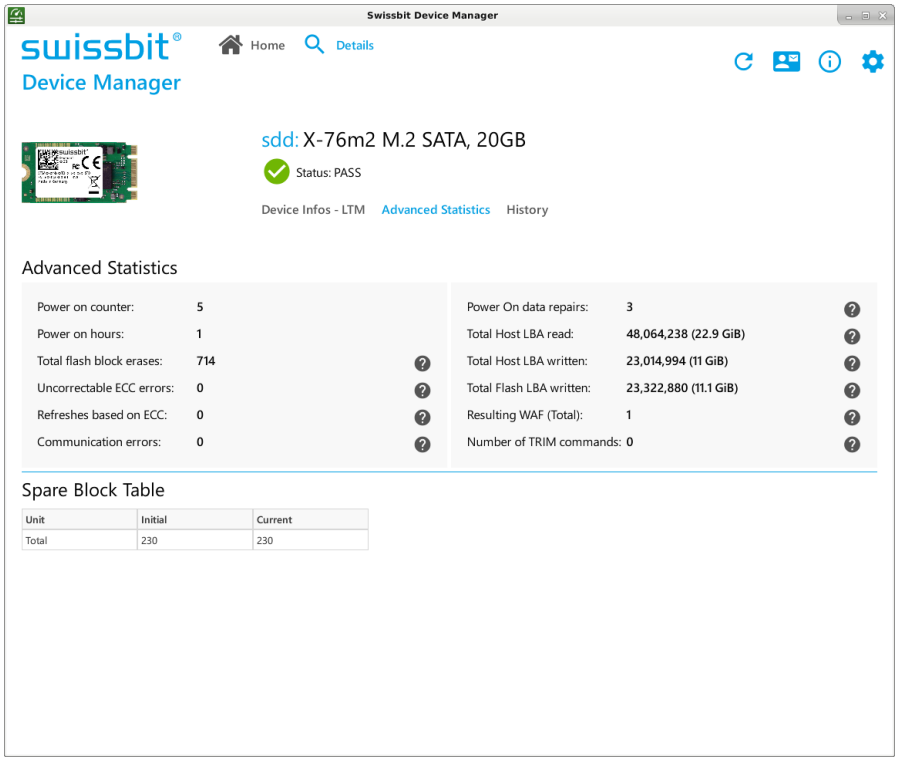
### 3.4.1 Device Information

The Device Information page provides an overview about the most important lifetime information. The page shows a status for all relevant properties of the device. The status can be “PASS”, “WARNING” or “FAIL”.



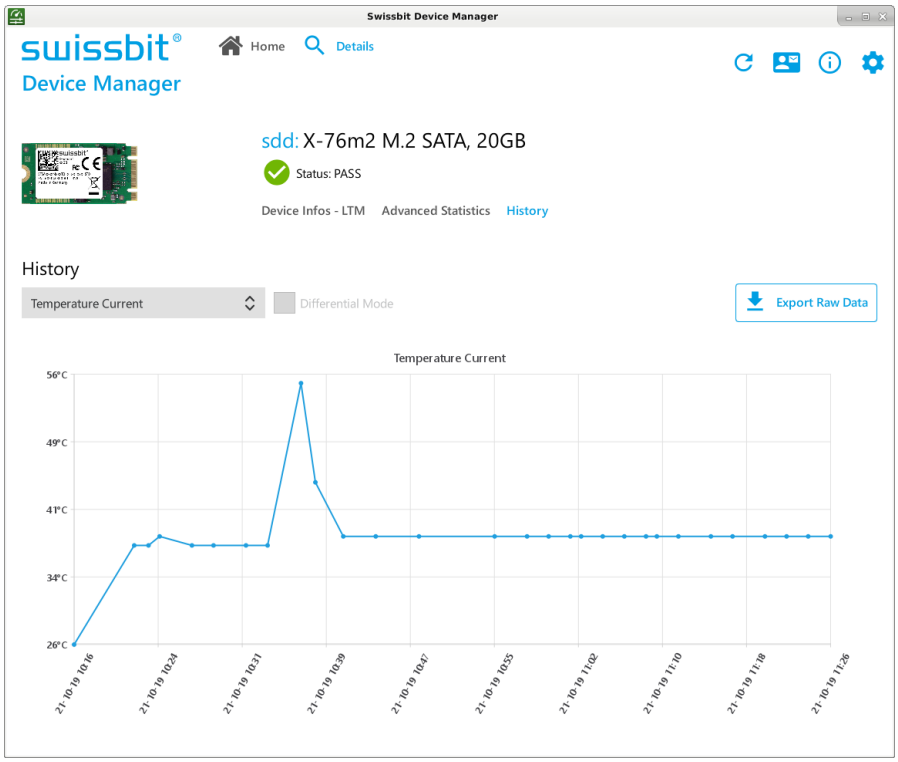
### 3.4.2 Advanced Statistics

The Advanced Statistics shows even more details about the device:



### 3.4.3 Device History Charts

On the History page the progress of all values can be displayed on a chart:




This displays all samples that have been taken from the device. The lower axis designates the sample time, the left axis the value. Please note that the chart is only representative with multiple samples (several hours or days). To see how much the value has changed over time, select the "Differential Mode".

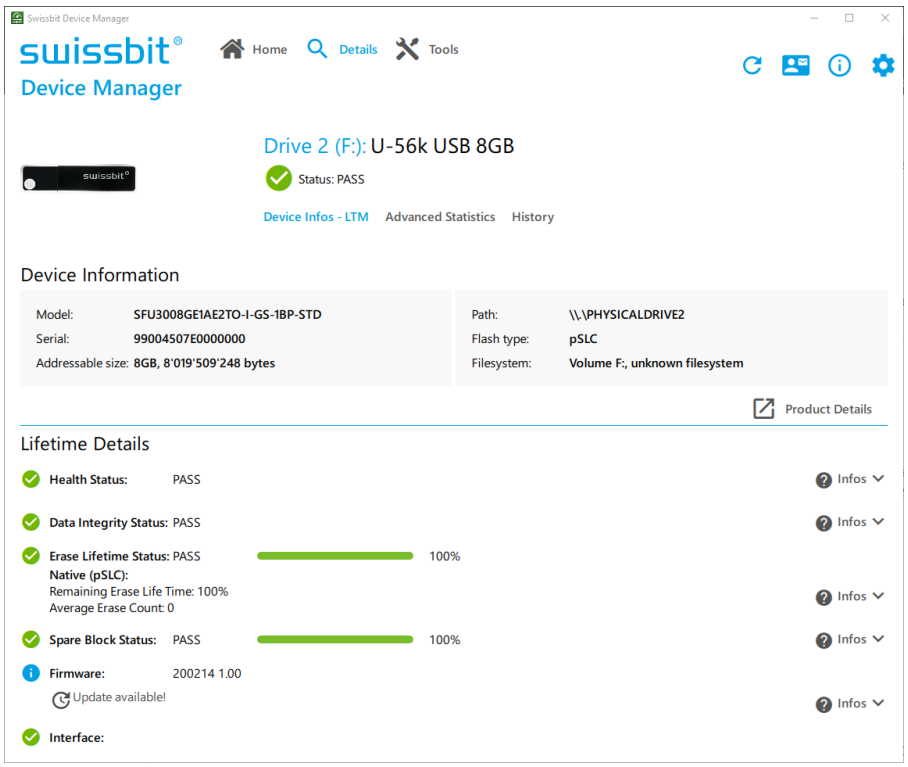
To export the logged data of the selected drive as XML or CSV file, please select "Export raw data".

## 3.5 Tools: Firmware Update

In the Tools section of the utility, the SBDM tool can apply firmware updates for many Swissbit devices.

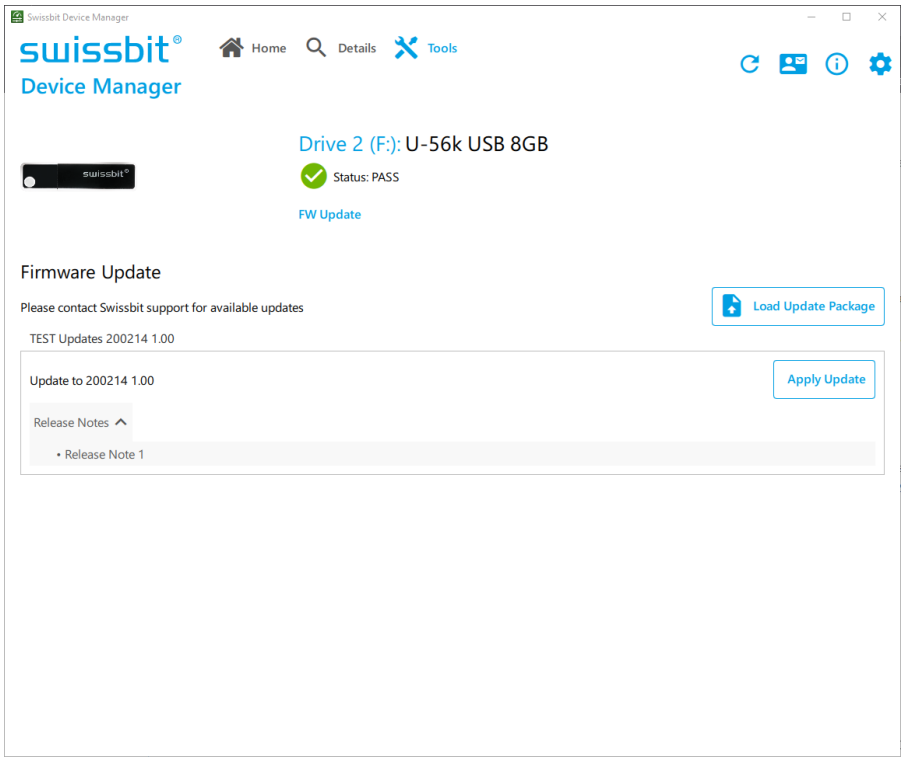
Firmware update packages are provided by Swissbit technical support on request only, as files with "sbdm" extension. Packages are automatically detected if placed in the working directory of the utility (near sbdm.db file). Additional packages from a different folder can be manually loaded by the user in the Tools section with the button "Load Update Package".

If an update is found to be available, the Home and Details pages will highlight firmware availability.



### 3.5.1 Performing an update

Please **back up device data** before performing a firmware update! Swissbit cannot be held responsible for any data loss incurred by a failed firmware update.




The Apply Update button will perform the device update and will report status information.

# 4 Usage of command line utility

The command line utility is provided for users that prefer command line or that would like to integrate the tool into the infrastructure. The tool output is mostly compatible with the older SBLTM utility.

## 4.1 Launching SBDM

### 4.1.1 Windows

To start the application, launch sbdm-cli.exe from a console with administrative privileges. The console can be started by starting cmd.exe "as Administrator" (i.e. Ctrl + Shift + Enter in start menu).

### 4.1.2 Linux

To start the application:

3. Make executable with: "chmod a+x sbdm-cli
4. Start as root or with sudo: "sudo ./sbdm-cli"

### 4.1.3 Command line arguments

The utility will print usage information with –h or --help

```
Usage: ./sbdm-cli [options] [mode] [device]
Options:
  -h, --help, -?                  Displays help on commandline options.
  -v, --version                   Displays version information.
  -q, --quiet                     No textual output except errors. For
                                  firmware update, this enforces unattended
                                  mode (read-only).
  -o, --output, -l, --list <list> Information output mode, can be "human"
                                  (default) or "csv". Supported for LTM mode
                                  only.
  -f, --fw <file>                 Firmware package to use for firmware update.
                                  If this option is omitted, the tool will
                                  search firmware package automatically. Can be
                                  provided multiple times for multiple
                                  packages.
  --db <db>                       LTM mode only, sets database sbdm.db file
                                  usage, can be "auto", "no" or "yes".

Arguments:
  mode                            Mode to use. Available are "ltm" to get
                                  device data and "fwupdate" for firmware
                                  update.
                                  "fwupdate-force" is an unattended mode with
                                  forced update execution, "fwupdate-query" is
                                  an unattended mode to query for possible
                                  updates only.
                                  Unknown keywords will be treated as [device]
                                  arguments in mode "ltm".
  device                          Device to use. Provide the full device path
                                  (i.e. /dev/sdx).
```

If only a specific drive needs to be checked or updated, the command line allows providing the device path on the command line after the "ltm" or "fwupdate" keywords.

- Windows: provide a drive number as assigned by Microsoft Windows or the full path (i.e. \\.\PHYSICALDRIVE1, case sensitive).
- Linux: provide a device path, i.e. /dev/sdx

## 4.2 LTM mode

By default or when providing the "ltm" keyword on the command line, the utility is running in LTM mode, and returns device status information in a human readable form.

Example output:

```
Swissbit Device Manager 1.2.0

/dev/sdb
--------

  Device information
    Model:                        SFU3064GC2AE2TO-I-LF-1A1-STD
    Series:                       U-50n
    Serial:                       601708389200004C
    Firmware:                     200214 1.00
    Capacity:                     64GB
    Overall status:               PASS
    Health status:                PASS
    Data integrity status:        PASS

  Spare block information
    Spare block status:           PASS
    Remaining spare blocks:       100%
    Current spare blocks (total): 1414
    Initial spare blocks (total): 1414

  Erase count information
    Erase status:                 PASS
    Remaining erase life time:    100%
    Flash cell mode:              Native
    Average erase count:          4
    Total erase count:            70680

(...)

  Interface
    Interface status:             PASS
    Current speed:                SuperSpeed (5Gbps)
    Max. supported speed:         SuperSpeed (5Gbps)
    Interface errors:             0
```

The output can be changed to comma separated values (CSV) by providing the parameter "--output csv".
Example output:

```
Path;Model;Series;Serial;Firmware;Capacity LBA;Overall status;Health status;(...)
/dev/sdb;SFU3064GC2AE2TO-I-LF-1A1-STD;U-50n;601708389200004C;(...)
/dev/sdc;SD card SB AFLI1;S-58;0012345686;(...)
```

## 4.3 Firmware update mode

When providing the "fwupdate" keyword or its extensions on the command line, the utility is running in Firmware Update mode. The following modes are available:

- fwupdate: interactive firmware update on console
- fwupdate-query: non-interactive mode to check availability of firmware updates
- fwupdate-force: non-interactive mode that applies all available firmware updates without confirmation

Example output:

```
Swissbit Device Manager 1.2.0

/dev/sdb
--------

  Device:
    Model:             SFU3064GC2AE2TO-I-LF-1A1-STD
    Series:            U-50n
    Serial:            601708389200004C
    Current firmware:  200214 1.00
    Updates available: yes

  Update to 200214 2.00 T1
     Release notes:  - Release Note 1

  >>> Please backup the data on the device before performing an update!

  Would you like to update this device?
  Enter yes or y to update the device (n or no to continue to next device):
```

Firmware update packages are searched in the working directory by default. Specific packages can be provided on the command line with the "--fw" parameter, in this case the working directory is not searched.

## 4.4 Return codes

The utility returns the following status codes to indicate success or failure:

**Table 1: Return codes**

| Return code | Meaning |
|---|---|
| 0 | Success<br>LTM mode: all devices have PASS status<br>Firmware update query mode: no updates available<br>Firmware update force mode: updates applied |
| 1 | Internal error (please report to Swissbit) |
| 2 | Failed enumeration (possible issue with access privileges) |
| 3 | LTM mode: No supported devices<br>Firmware update: no updates found for any device |
| 4 | Failed to read information from device (possible issue with access privileges) |
| 5 | Generic failure<br>LTM mode: a drive reports FAIIL status<br>Firmware update: update failed |
| 6 | Some devices requested on command line were not found or are not supported |

# 5 Firmware Update Linux Live CD manual

For systems that use other operating systems or have limited user access rights, Swissbit can provide Linux Live packages for convenient firmware update. These packages include a minimal x86 Linux operating system which can be booted from a USB stick or a CDROM.

The Linux live image supports legacy and UEFI boot.

## 5.1 Creating a bootable medium

The firmware update package is provided as a bootable ISO 9660 image.

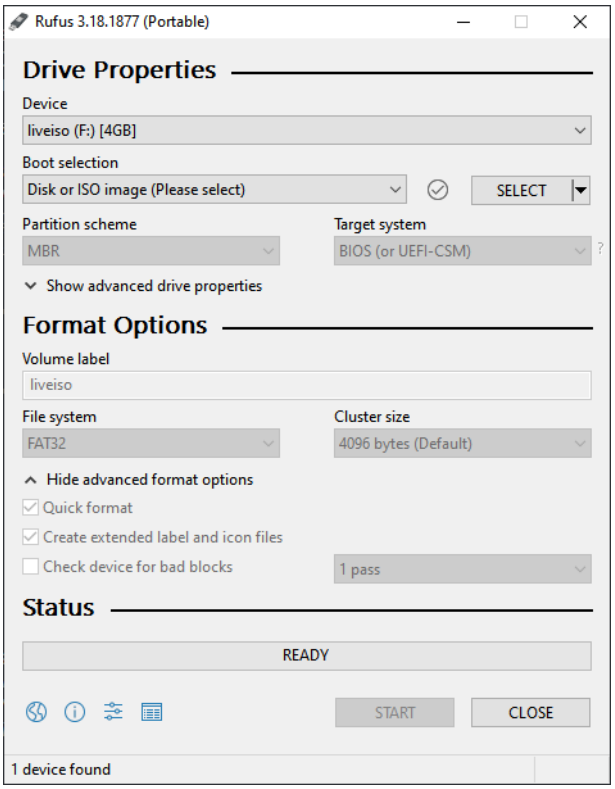
It can be used to generate a bootable USB stick, or a bootable CD.

## 5.2 Bootable USB stick

Requirements: empty USB stick with size 32MB or more. There are different tools to build a bootable USB stick from the ISO file, for Windows Swissbit recommends Rufus or core2usb.

Steps to build with Rufus on Windows:

- Download and open the latest Rufus release
- Select the target drive in the device list
- Use SELECT button to choose the ISO file provided by Swissbit.
  The other settings are automatically adjusted after this selection.




- For older systems, it might be required to select the checkbox "Add fixes for old BIOses" under advanced device properties.
- **Make sure to write the image in dd mode (not ISO).**

## 5.3 Update host requirements

The card update host (PC) where the update is to be performed needs be able to boot from the boot media (see previous chapter). An Intel i486 compatible CPU and minimum of 128MB RAM is required.

Hot swapping (removing and plugging in) is supported for SATA devices (CFast / SSD) only when the chipset is configured in "AHCI" or "Native" mode. After plugging in the device, wait a few seconds and press "Enter" to refresh the device list.

## 5.4 Usage

- Please first **back up the data** on the target device!
  Swissbit cannot be held responsible for any data loss incurred by a failed firmware update.
- Boot a system from the medium created earlier (see chapter 5.1 Creating a bootable medium). Often the BIOS will need to be configured to boot from USB or CDROM, consult your system manual for details.
- Wait until the boot process is complete and the drive scan was executed. See chapter 4.3 for an example output

  If a device is not recognized by the tool, please check the BIOS for the SATA chipset settings. When using the "compatibility" or "IDE" mode, the device should be detected ons any mainboard.
- Follow the instructions on the screen

# 6 Troubleshooting

## 6.1 Connecting a device indirectly (via bridge chip)

SBDM officially only supports bridge chips listed in this chapter. If a bridge chip (adapter) is used that is not supported the device might be not listed by SBDM at all or not all SBDM features might be supported.

### 6.1.1 NVMe

SBDM currently supports the following USB-NVMe bridge chips:

- JMicron JMS583
- ASMedia ASM236x

### 6.1.2 SD & eMMC

SBDM currently supports the following connections for Swissbit SD & eMMC series:

**Table 2: Connections SD & eMMC**

| Method | S-200 / S-220(u) | S-4x(u) | S-5x(u) | EM-2x | EM-3x |
|---|---|---|---|---|---|
| Device connected directly to chipset | Yes (Linux only) | Yes | Yes | Yes (Linux only) | Yes (Linux only) |
| Device connected indirectly (Via USB-SD or USB-eMMC Reader) | No | Yes | Yes | No | No |

### 6.1.3 CompactFlash

SBDM uses the S.M.A.R.T. standard to read out the lifetime information out of Swissbit CF Cards. USB card readers are quite difficult, because the USB Protocol does not natively support S.M.A.R.T. commands. They basically transfer only standard read and write commands, but do not forward the SMART commands to the card. There are some USB to IDE chipsets which do support SMART over special USB commands. There are several special standards/commands to transfer SMART over USB:

- SCSI/ATA Translation SAT(12)
- SCSI/ATA Translation SAT(16)
- Manufacturer specific: Cypress
- Manufacturer specific: JMicron

SBDM does support all of them and does try to detect if the reader does support one of them. To read out lifetime values, the card reader has to support one of those.

Unfortunately, we do not know about any USB->CF reader supporting one of these commands. Usually, "hot-plug-capable" CF/USB-adapters do not transfer the SMART information but traditional USB to IDE readers mostly do. These "not-hot-plug-capable" IDE adapters are often used to connect traditional IDE-HDDs to USB. These readers have an IDE-plug (ATAPI-40). Therefore, an IDE to CF connector board is needed additional to the reader to connect the CF card. PCMCIA cards slots (often built in a Laptop/Notebook) with a CF to PCMCIA adaptor work as well (Pc card mode).

USB/IDE-Adaptor/Reader recommendation: One approved example of a USB/IDE-adaptor is the Sharkoon DriveLink IDE/SATA to USB 2.0 adaptor: http://en.sharkoon.com/product/1686/06403

Attention: There is also a USB3 reader available from Sharkoon, but this one does not support SMART over USB!

Additional, an IDE to CF Adaptor-board is needed. It is important that the board is DMA capable. One of these DMA capable IDE/CF-Boards it the CFDISK.1E. from PC Engines: http://pcengines.ch/cf1eh.htm

Alternatively, you can also use any standard motherboard (with an IDE Interface) and connect the card via IDE to CF Adaptor to the IDE port. The disadvantage of this is that you will have to boot the system with the card attached (IDE is not hot-plug-capable).

Another solution is to connect the CF card to a SATA port of your motherboard (CF -> IDE to CF Adaptor -> IDE to SATA adapter). This solution usually also supports hot-plug.

### 6.1.4 SATA / CFast

SATA-USB bridge chips with SCSI/ATA Translation (SAT) are supported by SBDM. Please see chapter 6.1.3 for details.

## 6.2 No drives detected on Linux

Please make sure SBDM is launched with root privileges and that the SBDM-<arch>.AppImage application file has execution rights (chmod +x ./SBDM-<arch>.AppImage)

## 6.3 No SMART data on Windows for IDE / SATA

If the IDE / SATA chipset of the PC uses special drivers, it may not be possible to read the SMART information. Not all driver manufacturers support SMART using standardized interfaces. Best support is to be expected with the default Microsoft drivers or Intel chipset drivers.

## 6.4 SD card with write protection feature not detected

S-4x family products allow reading out life time monitoring values over generic USB card readers.
There are some limitations with USB card reader support.

The SBDM tools requires write access to the SD card for this functionality. Write protected cards are not guaranteed to be supported.

- Hardware write protection (by hardware switch) will never work (with any card reader)
- Software write protection by SD Temporary Write Protect or Permanent Write Protect will work with certain card readers only. Swissbit has successfully tested the following card readers:
  - SanDisk MicroMate SDDR-113 (recommendation)
  - Kingston FCR-HS219/1

## 6.5 Application is not properly rendered and very slow on embedded devices

Please make sure OpenGL drivers are enabled so hardware acceleration can be used. On embedded devices the CPU might not be powerful enough to render the SBDM application.

### 6.5.1 Raspberry Pi

For Raspberry Pi make sure to enable GL Driver.

1. Open the Raspberry Pi Software Configuration Tool with "sudo raspi-config"
2. Go to "Advanced Options"
3. Go to "GL Driver"
4. Enable "G1 GL (Full KMS)"
5. Reboot

# 7 License

SBDM is distributed with the file licenses.txt, which contains the license text, including open source licenses used.

# 8 SBDM Revision History

Table 3: SBDM Revision History

| Date | Revision | Changes |
|---|---|---|
| 2022-02-21 | 1.0.0 | ○ Initial Release based on feature set of SBLTM 1.7.2 |
| 2022-08-16 | 1.1.0 | ○ Add support for S-6x and N-3xm2 series |
| 2022-10-03 | 1.1.1 | ○ Improve stability for S-5x family to avoid seldom corrupted readouts<br>○ Fix cases of corruption of mounted file systems, which was observed if no partition table was used (Linux only)<br>○ Improve stability for N-3x LTM readout (Windows issue workaround)<br>○ Show selected speeds for SATA drives |
| 2023-04-18 | 1.2.0 | ○ Add command line interface utility<br>○ Add support for Firmware Update functionality<br>○ Add Windows x64 binaries<br>○ Add support for N4x and N5x series<br>○ Add temperature sensor support for S-6x family<br>○ Change Interface Speed warning to an informational notice<br>○ Improve responsibility on busy devices<br>○ Improved compatibility with some graphics drivers and specific non-Swissbit devices |
| 2023-11-15 | 1.2.1 | ○ Add firmware update support for Intel RST RAID for Windows<br>○ Add minimize option for GUI mode<br>○ Fix application crash for G2x series<br>○ Add support for D1x series |
| 2024-03-07 | 1.2.2 | ○ Improved Intel RST RAID compatibility by also showing failed RAID drives<br>○ Improved stability during refresh<br>○ Improved FW update compatibility<br>○ Improved CLI compatibility on Linux by using AppImage<br>○ Show busy indicator while loading FW update packages<br>○ Added series decoding for security products (PS-6x and PU-50n) |
| 2024-04-10 | 1.2.3 | ○ Add support for F-8x & X-8x series with custom model number<br>○ Fix not shown firmware revision for U-46 series |
| 2024-05-02 | 1.2.4 | ○ Improved FW update compatiblity for USB products<br>○ Fixed invalid "Flash LBA Written" counter for X-5x series |
| 2026-03-12 | 1.3.0 | ○ Adding support for new series M1x, U1x, A2x, N7x<br>○ Multiple small improvements |

# 9 Document Revision History

Table 4: Document Revision History

| Date | Revision | Changes | Details |
|---|---|---|---|
| 2022-02-08 | 1.00 | ○ First Release | – |
| 2022-08-16 | 1.01 | ○ Update supported series<br>○ Update SBDM revision history | – |
| 2022-10-06 | 1.02 | ○ Update SBDM revision history | – |
| 2023-05-02 | 1.03 | ○ Update SBDM revision history<br>○ Add documentation for command line executables and new package format | Doc. req. no. 6225 |
| 2023-06-30 | 1.04 | ○ Update SBDM supported product lines<br>○ Corrected typos | Doc. req. no. 6258 |
| 2023-08-16 | 1.05 | ○ Add Firmware Update Linux Live CD chapter | Doc. req. no. 6666 |
| 2023-11-15 | 1.06 | ○ Update SBDM revision history<br>○ Add Command Line arguments chapter for GUI mode | Doc. req. no. 6966 |
| 2024-03-07 | 1.07 | ○ Update SBDM revision history | Doc. req. no. 6966 |
| 2024-04-10 | 1.08 | ○ Update SBDM revision history | Doc. req. no. 7016 |
| 2024-05-02 | 1.09 | ○ Update SBDM revision history | Doc. req. no. 7062 |
| 2026-03-12 | 1.10 | ○ Update Live Linux chapter<br>○ Update SBDM supported product lines<br>○ Update SBDM revision history | Doc. req. no. 7782 |