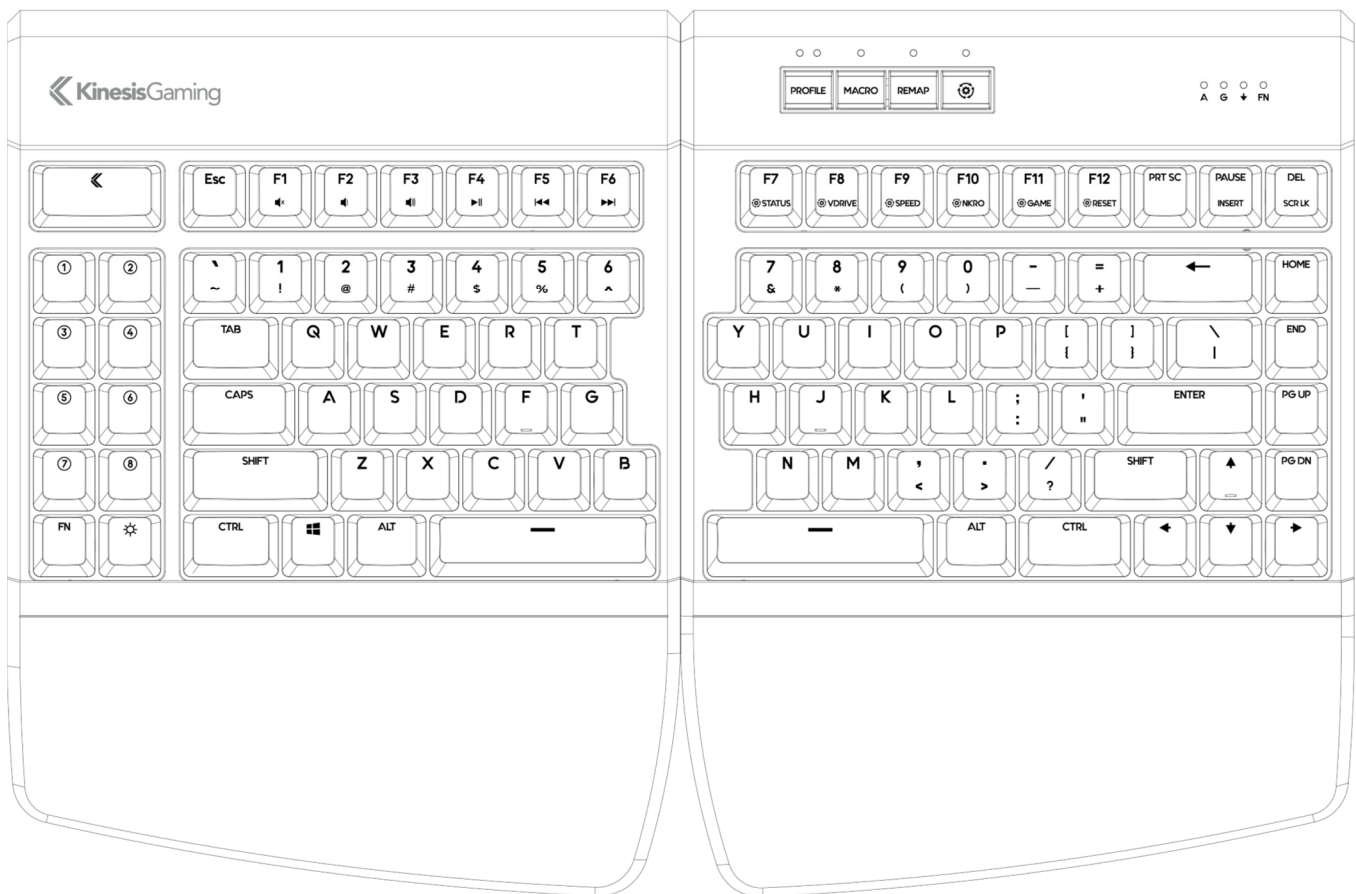




FREESTYLE EDGE

RGB

SPLIT MECHANICAL GAMING KEYBOARD



KB975 Direct Programming Guide

Keyboard models covered by this manual:

- KB975-BLU
- KB975-RED
- KB975-BRN
- KB975-SIL

www.KinesisGaming.com

Version 1.0.86 (August 15, 2019)

This manual covers features included through firmware version 1.0.0. To download the latest firmware and to access all support resources visit:

www.KinesisGaming.com/Support.

To shop for accessories visit:

- Lift Kit- Adjustable tenting accessory (5°/10°/15°) for use with Palm Supports
- V3 Pro- Adjustable tenting accessory (5°/10°/15°) for use without Palm Supports
- XL Mouse Pad- Oversized Desk Mat/Mousepad designed for use with the Freestyle Edge
- Vektor– RGB Gaming Mouse (Dual-Zone RGB Lighting, 6 Programmable Buttons)

KinesisGaming.com/shop

© 2019 by Kinesis Corporation, all rights reserved. Kinesis and Freestyle are registered trademarks of Kinesis Corporation. Freestyle Edge, SmartSet, and v-Drive are trademarks of Kinesis Corporation. All other trademarks are property of their respective owners.

Information in this document is subject to change without notice. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any commercial purpose, without the express written permission of Kinesis Corporation.

FCC Radio Frequency Interference Statement

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help

Warning

To assure continued FCC compliance, the user must use only shielded interfacing cables when connecting to computer or peripheral. Also, any unauthorized changes or modifications to this equipment would void the user's authority to operate.

1.0 Introduction

The Freestyle Edge RGB is a fully-programmable keyboard that does not use any special drivers or software so it is compatible with all mainstream USB operating systems. The keyboard was designed to be programmed quickly and easily using the onboard programming tools (all operating systems) and/or the powerful SmartSet App (for Windows and Mac). However, power users can also Direct Program the keyboard on any operating system, which is the topic of this guide.

2.0 Direct Programming Overview

The Freestyle Edge RGB has 9 customizable Profiles, which comprise 9 sets of layouts and lighting configurations. The keyboard also has a series of Global Keyboard Settings which can be configured.

Each of these configurations are stored on the keyboard's virtual flash drive (called the "v-Drive") as a series of simple text files (.txt), which can be accessed in Windows, Linux, Mac, and Chrome. Each time a remap or macro is created, it is written to the corresponding layout .txt file as a discrete line of "code". Each time a Lighting Effect is assigned to a Profile or customized in the App, the parameters are written to a corresponding led.txt file. Each time a keyboard setting is changed, the change is recorded in the appropriate field in the "kbd_settings.txt" file. With direct programming, power users can bypass the onboard programming tools and the SmartSet App to edit these files directly to customize their keyboard.

3.0 Before you Begin

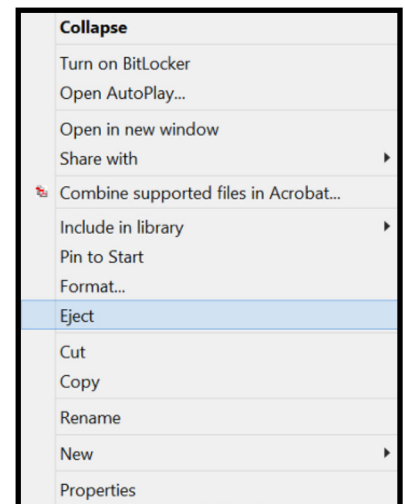
3.1 Power Users ONLY

Direct editing requires learning to read and write a custom syntax. The insertion of incorrect characters into any of the configuration files can have unintended consequences and could cause temporary problems with even basic keyboard operation. Read the Quick Start Guide and User Manual first and proceed with caution.

3.2 Always Eject the v-Drive before disconnecting the v-Drive

After editing any .txt files on the v-Drive, it is necessary to first save and close the files, and then use the appropriate eject protocol for your operating system *before* closing the v-Drive with the onboard shortcut. The eject protocol ensures that your PC is not actively reading/writing to the v-Drive and will prevent the type of data corruption that can occur and any conventional "thumb" flash drive if it is disconnected unexpectedly.

Windows Eject: Save and close any .txt files you have been editing. From File Explorer, navigate back to the top level of "FS EDGE RGB" removable drive and right click the drive name and then select Eject. Once you receive the "Safe to Eject" notification you may proceed to closing the v-Drive with the onboard shortcut. Failure to eject can result in a minor drive error that Windows will ask you to repair (see Section 6.7).



3.3 Non-US Users

Your computer must be configured for the English (US) keyboard layout. Other language drivers use different codes/positions for certain keys which are critical for programming characters such as `[]`, `{}` and `>`.

3.4 Simple Text Files ONLY

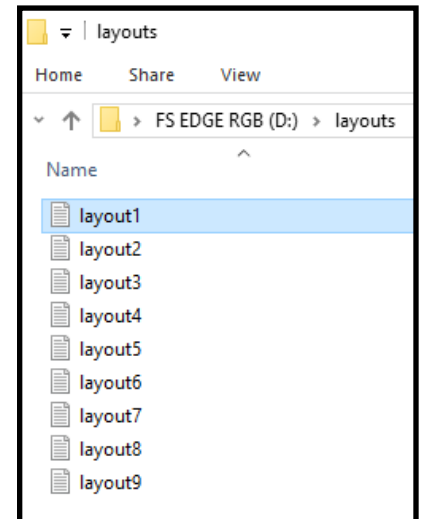
Do not save configuration files in the Rich Text Format (.rft) as special characters could lead to errors.

4.0 Direct Programming Layouts

The Edge features 9 configurable Profiles, each with its own corresponding “layout” (1-9). The nine default layouts are saved as separate .txt files in the “layouts” subfolder on the v-Drive. Only custom remaps and macros are saved to the file, so if no changes have been made to a layout, the file will be empty. Users can either write code from scratch or edit keyboard-generated code using the syntax rules described below. Deleting a layout file will permanently delete its stored remaps & macros, but the keyboard will automatically regenerate a blank layout file.

4.1 File Naming Convention

Only the nine numbered layouts can be loaded to the Freestyle Edge RGB. Additional “backup” layouts can be saved as .txt files with descriptive names, but they cannot be loaded to the keyboard without renaming them first.



4.2 Syntax Overview– Position & Action Tokens

Remaps and macros are encoded in a layout file using a basic programming syntax. Each of the 95 full-size keys on the keyboard has been assigned a unique “Position” token used to identify that key for programming.

Each keyboard & mouse action supported by the Edge has been assigned a unique “Action” token corresponding to a standard USB “scan code”.

To successfully re-program a key, each line of code must include a position token and one or more action tokens. The “>” symbol is used to separate position tokens from actions tokens, and individual tokens are surrounded by brackets. Examples:

- Remaps are encoded with Square Brackets: [position]>[action]
- Macros are encoded with Curly Brackets: {trigger key position}{modifier co-trigger}>{action1}{action2}...

4.3 Layout Programming Tips

- If the keyboard cannot understand the desired remap, then the default action will remain in effect.
- Do not mix and match square and curly brackets in a single line of code
- Separate each line of code with Enter/Return
- The order in which the lines of code appears in the .txt file does not generally matter, except in the event of conflicting commands, in which case the command closest to the bottom of the file will be implemented.
- Tokens are not case-sensitive. Capitalizing a token will not produce the “shifted” action (See Section 6.5).
- A line of code can be temporarily disabled by placing an asterisk (*) at the beginning of the line.

4.4 Position Tokens

Generally speaking, position tokens are defined by the basic QWERTY Windows action for the key in the default layout. In some cases tokens have been modified for clarity and/or ease of programming. Each physical key has two tokens, one for the top layer action and one for virtual embedded layer action. To designate a key position in the Fn layer, you simply encode the prefix “fn” (with a space) before the basic, top-layer token, and outside of the brackets.

- Example: The embedded Hotkey 1 position is: `fn [hk1]>...`

However, for the eight keys which have a unique embedded key action in the default layout, there is a unique Fn layer token which corresponds to the embedded layer action for that key position.

4.5 Position Token Map



4.6 Programming Remaps

To program a remap, encode the position token and the action token in square brackets, separated by ">".

Remap Examples:

1. Hotkey 1 performs Q: `[hk1]>[q]`
2. Escape key performs Caps Lock: `[esc]>[caps]`
3. The Windows key in the Fn layer performs Right Shift: `fn [lwin]>[rshft]`

Programming the Fn Key: The Edge RGB supports two different types of Fn layer access: "Shift" and "Toggle". The default behavior of the Fn key is now "Shift" (i.e., momentary). *Note: For proper functionality, it is necessary to assign the desired Fn action token to the position token in both layers, otherwise you'll get stuck in the Fn layer*

4.7 Programming Macros

To program a macro, encode the position token(s) and the action token in curly brackets, separated by ">". Each layout can store up to 7,200 total macro characters spread across up to 100 macros.

Trigger Keys: Any non-modifier key can be assigned as a macro trigger in either layer. A co-trigger can be added by adding a modifier key to the left of ">". *Note: Windows keys are not recommended as co-triggers.*

Shifted Actions: To produce a shifted key action, it is necessary to encode a macro which includes the both the down and up stroke of the shift key surrounding the basic key action. Downstrokes are indicated by placing a "-" inside the bracket and upstrokes are indicated by placing "+". See example 1 below.

Individual Playback Speed Prefix {s_}: By default, all macros play at the selected "Global" playback speed. To assign a custom speed for improved playback performance for a given macro you can use the "Individual Playback Speed" prefix "{s_}". Choose a number from 1-9 corresponding to the speed scale shown Section 4.6. The speed prefix should be placed to the right of the ">" before the macro content. See example 2 below.

Multiplay Prefix {x_}: By default, all macros playback continuously while the trigger key is held. To override the repeat feature and restrict a macro to playback a specific number of times you can use the "Macro Multiplay" prefix "{x_}". Choose a number from 1-9 corresponding to the number of times you want the macro to replay. The multiplay prefix should be placed to the right of the ">" before the macro content. See example 3 below.

If a macro is not playing back properly, try assigning a Multiplay value of 1. The macro may actually be firing multiple times before you are releasing the trigger key.

Timing Delays: Delays can be inserted into a macro to improve playback performance or to produce a mouse double-click. Delays are available in any interval between 1 and 999 millisecond ({d001} & {d999}), including random delays ({dran}). Delay tokens can be combined to produce delays of various durations and inserted at any point in a macro. See example 4 below.

Macro Examples:

1. Pause key performs "Hello" with a capital H: `{pause}>{-lshft}{h}{+lshft}{e}{l}{l}{o}`
2. Hotkey 5 + Left Ctrl performs "qwerty" at speed 9: `{lctrl}{hk5}>{s9}{q}{w}{e}{r}{t}{y}`
3. Hotkey 1 increases the volume 3 notches: `{hk1}>{x3}{vol+}`
4. Embedded Tab key performs one left-mouse double click: `fn {tab}>{x1}{lmous}{d125}{lmous}`

4.8 Tap and Hold Actions (*Firmware update required).

With Tap and Hold, you can assign two unique actions to a single key based on the duration of the keypress. Each Layout can support up to 10 Tap and Hold actions. Designate the position token, then the Tap action,

then the timing delay from 1 to 999 milliseconds using the special Tap and Hold token ({t&hxxx}), then the Hold Action. Due to inherent timing delays, Tap-and-Hold is not recommended for use with alphanumeric typing keys. Not all key actions support Tap-and-Hold.

Note: For most applications, we recommend a timing delay of 250ms.

Tap and Hold Example:

- Caps performs Caps when tapped and Esc when held longer than 500ms: [caps]>[caps][t&h500][esc]

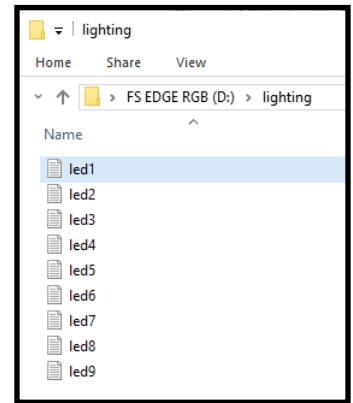
4.8 Action Token Dictionary (*Firmware update required)

A	B	C	D	E	F	G	H	I	J	K	L
M	N	O	P	Q	R	S	T	U	V	W	X
Y	Z	1	2	3	4	5	6	7	8	9	0
HYPH	=										
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24
Left Shift		Right Shift		Left Alt		Right Alt		Left Windows & Command		Right Windows & Command	
LSHFT		RSHFT		LALT		RALT		LWIN		RWIN	
Left Ctrl		Right Ctrl									
LCTRL		RCTRL									
Left Windows + Left Ctrl + Left Alt + Left Shift *						Left Ctrl + Left Alt + Left Shift *					
HYPER						MEH					
Open Bracket		Close Bracket		Period		Comma		Apostrophe		Tilde	
OBRK		CBRK		PER		COM		APOS		TILDE	
Forward Slash		Back Slash									
/		\									
Mute	Volume Up	Volume Down	Play/Pause	Next Track	Previous Track	Left Mouse	Right Mouse	Middle Mouse	Mouse Button 4	Mouse Button 5	
MUTE	VOL+	VOL-	PLAY	NEXT	PREV	LMOUS	RMOUS	MMOUS	MOUS4	MOUS5	
Enter		Tab		Space		Delete		Backspace		Home	
ENT		TAB		SPC		DEL		BSPC		HOME	
End		Page Up		Page Down							
END		PUP		PDN							
Left Arrow		Right Arrow		Up Arrow		Down Arrow		Escape		Print Screen	
LFT		RGHT		UP		DWN		ESC		PRNT	
Pause		Scroll Lock		Insert							
PAUSE		SCRLK		INS							
Num Lock		Keypad 1		Keypad 2		Keypad 3		Keypad 4		Keypad 5	
NUMLK		KP1		KP2		KP3		KP4		KP5	
Keypad 6		Keypad 7		Keypad 8							
KP6		KP7		KP8							
Keypad 9		Keypad 0		Keypad Plus		Keypad Minus		Keypad Divide		Keypad Multiply	
KP9		KP0		KP+		KP-		KP/		KP*	
Keypad Enter		Keypad Equals		Keypad Decimal							
KPENT		KP=		KP.							
Caps Lock		Fn Toggle		Fn Shift		Menu/App		International		Calculator	
CAPS		FNTOG		FNSHF		MENU		INTL/		CALC	
Shutdown		LED Toggle		Macro Delay		Random Delay		No Key Action			
SHTDN		LED		Dxxx		DRAN		NULL			

5.0 Direct Programming Lighting Effects

The Edge features 9 configurable Profiles, each with its own corresponding “lighting effect” (1-9). The nine default lighting effects are saved as separate .txt files in the “lighting” subfolder on the v-Drive. If the file is blank, the keyboard lighting will be disabled.

Users can either write code from scratch or edit keyboard-generated code using the syntax rules described below. Deleting an led file will permanently delete the customizations for that effect, but the keyboard will automatically regenerate a the original effect for that Profile.



5.1 Keyboard Effects

There are a number of customizable keyboard level effects to choose from. Each Effect has one or more parameters that can be customized. Basic Syntax: [effect]>[parameter1][parameter2][parameter 3]

5.2 Keyboard Level Effect Tokens & Parameters

Monochrome: [mono] + Color

Breathe: [breathe]> [Speed] + Freestyle (see below)

Wave: [wave]>[Speed][Direction]

Spectrum: [spectrum]>[Speed]

Reactive: [reactive]>[Color][Speed]

Starlight: [star]>[Color][Speed]

Rebound: [rebound]>[Direction][Speed]

Loop: [loop]>[Color][Speed][Direction]

Pulse: [pulse]>[Speed]

Rain: [rain]>[Color][Speed]

Pitch Black: [black]

5.3 Parameters

Color: When applicable, the Color parameter is defined by inputting Red, Green, and Blue channel values inside square brackets (0-255): [RRR][GGG][BBB]

Examples:

1. Monochrome Solid Green: [mono]>[0][255][0]

2. Monochrome Solid Yellow: [mono]>[255][255][0]

Direction: When applicable, the Direction parameter is defined by inputting one of the following tokens: [dirup], [dirdown], [dirleft], [dirright]

Examples:

1. Wave Left Speed 5: [wave]>[spd5][dirleft]

Speed: When applicable, the Speed parameter is defined by inputting one a number 1-9 inside the speed token: (spdx) [dirdown], [dirleft], [dirright]

Examples:

1. Pulse Speed 5: [pulse]>[spd9]

5.4 Freestyle and Breathe Effects

The Freestyle and Breathe effects allow you to assign a custom to color to each of the 95 keys using the standard key position token and the color parameter. With Freestyle mode, there is now effect parameter, simply define the color values for each key. Keys not specified will be off. For Breathe, define the Breathe Effect first and specify the Speed parameter, then define the color values for each key.

Example 1: Set WASD keys to static Red

```
[w]>[255][0][0]
[a]>[255][0][0]
[s]>[255][0][0]
[d]>[255][0][0]
```

Example 1: Set Game Keys to Breathe at Speed 1 in Green

```
[breathe]>[spd1]
[hk0]>[0][255][0]
[hk1]>[0][255][0]
[hk2]>[0][255][0]
[hk3]>[0][255][0]
[hk4]>[0][255][0]
[hk5]>[0][255][0]
[hk6]>[0][255][0]
[hk7]>[0][255][0]
[hk8]>[0][255][0]
[hk9]>[0][255][0]
[hk10]>[0][255][0]
```

5.5 NEW- Base Color (Rain, Loop, Rebound, Starlight & Reactive) (*Firmware update required)

Lighting Expansion Pack 1 allows users to assign a “base color” using the mono token to select lighting effects for a two-tone effect.

Example 1: Set Reactive Effect to Green at Speed 1 with a Base Color of Red

```
[reactive]>[0][255][0][spd1]
[mono]>[255][0][0]
```

5.6 NEW– Fn Layer Lighting Color (*Firmware update required)

Lighting Expansion Pack 1 allows users to a unique secondary lighting effect that is activated whenever the Fn layer is accessed. Program the Fn layer effect the same way you would the top layer effect, just use the “Fn prefix” described in Section 4.4 to modify the Effect or the key position token. Note: The position tokens for key actions and Freestyle lighting are identical except for these 8 keys:

Fn [f1] Fn [f2] Fn [f3] Fn [f4] Fn [f5] Fn [f6] Fn [pause] Fn [del]

Example 1: Set WASD keys to static Red in the top Layer and Green in the Fn Layer

[w]>[255][0][0]

[a]>[255][0][0]

[s]>[255][0][0]

[d]>[255][0][0]

fn [w]>[0][255][0]

fn [a]>[0][255][0]

fn [s]>[0][255][0]

fn [d]>[0][255][0]

6.0 Direct Programming Settings

6.6 Programming Settings

Keyboard Settings

The current global settings for the keyboard are saved in the “kbd_settings.txt” file in the “settings” sub-folder. Users can update these settings by modifying the fields in this .txt file and then using the Refresh shortcut (SmartSet + Layout) to implement them.

- profile_sync_mode: By setting this value to “OFF”, you can load different pairs of layout and lighting files
- startup_file: Edit the name of the layout.txt file to instruct the keyboard which Profile (or layout) to load
- Led_mode: Designate the active Lighting Effect when Profile Sync Mode is disabled.
- macro_speed: Change the Global Macro Speed with “1-9” for speed, or “0” to disable (default = 5)
- game_mode: Disable Game Mode with “OFF” or enable with “ON”
- nkro_mode: Disable NKRO Mode with “OFF” or enable with “ON”
- status_play_speed: Change the Status Report Playback speed (see Section 4.13) with “1-4” for speed, or “0” to disable (default = 3).
- program_key_lock: Disable Program Locking with “OFF” or enable with “ON”
- v-drive: Force the v-Drive to open automatically every time the keyboard is plugged in with “auto” or require manual opening with “manual”.

SmartSet App Settings

In the settings sub-folder you may also notice the “app_settings.txt” file. This file stores your saved preferences for in-App notifications when using the SmartSet App. “On” disables the notification and “Off” enables the notification.

6.7 v-Drive Repair

If you disconnect the v-Drive or unplug the keyboard while the PC was actively writing to one of the files (i.e., without ejecting), the next time you open the v-Drive on a Windows computer you may receive a harmless drive error warning. If you receive this error, follow the prompts to quickly “repair” the drive.

