

User Experience Guidelines

Livescribe™ Platform SDK
Version 1.5

Copyright and Trademarks

LIVESCRIBE, ECHO, PULSE, and PAPER REPLAY are trademarks or registered trademarks of Livescribe, Inc. Anoto is a trademark of Anoto Group AB. All other brand and product names are trademarks of their respective owners.

Copyright © 2007-2010 Livescribe Inc. All rights reserved.

UserExpGuidelines-PlatformSDK-1.5.0-REV-B

12/20/2010 10:57 AM

Table of Contents

Copyright and Trademarks.....	ii
About this Document.....	vii
General Functionality.....	1
Using the Menu	1
Double-Tap Center.....	1
Tap Right.....	2
Tap Left.....	3
Tap Up and Tap Down.....	4
Application Names.....	4
Application Menus	5
Standardized Menu Structure.....	5
Menu Behavior	9
Menu Graphics	9
Display	9
Display Area.....	9
Horizontal Scroll Speed.....	10

User Experience Guidelines

The OLED Must Never Be Blank	10
Data on Display	11
Providing Help	11
Help Menu Item.....	11
Audio Punctuation Marks.....	12
Application Deactivation.....	13
One-Second Deactivation.....	13
Livescribe Standard Controls.....	14
Do Not Repurpose Standard Controls	14
Open Paper	14
Termination Events.....	14
Timeout	15
Double-Tap on Open Paper.....	15
Special Characters.....	15
Prompt and Believe	16
Dynamic Active Regions.....	16
Shaping Dynamic Active Regions	17
Activating Functions	17
Paper Replay	18

User Experience Guidelines

Background Recording..... 18

Snapback..... 18

User Experience Guidelines

About this Document

This guide presents user interface guidelines for creating penlets to ensure a consistent user experience across all smartpen applications. Livescribe follows these guidelines for applications we develop in-house, such as Paper Replay, Paper Piano, and others.

General Functionality

Using the Menu

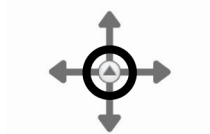
The Livescribe smartpen uses a flat 2D model for its menu system. The model is essentially a tree-like structure of menu items with submenu items. The best way to understand this model is to use the Nav Plus paper control to activate the Main Menu on your Livescribe smartpen. Then, experiment using the record and playback features of Paper Replay. Refer to the Paper Replay section of the *Livescribe Smartpen User Manual* for detailed instructions on navigating using the Nav Plus.

There are several basic Nav Plus actions reserved to control menus and applications:

- Double-Tap Center
- Tap-Right
- Tap-Left
- Tap-Up
- Tap-Down

Double-Tap Center

double-tap center

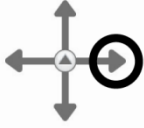


Double tapping on the center of the Nav Plus returns the display to the Main Menu and exits any currently-running application.

This action occurs no matter what application was active or what mode it was in, and thus this action is reserved and should not be assigned for any other actions.

Tap Right

tap right



A right-tap on the Nav Plus has different effects depending on the current application, mode, and state. In general, this is used to activate, select, launch, move forward, dive deeper, or confirm.

Examples include:

Launch

- Starts an application. For example, a right-tap from “Paper Replay” in the Main menu will launch the Paper Replay application.

Execute

- Executes or begins a function. For example, a right-tap from “Record new session” (in the Paper Replay app menu) will begin the recording of a new session.

Select

- Chooses an element to perform an operation on. For example, after selecting “Play session” in Paper Replay, the user can browse through a list of sessions; a right-tap begins playback of whichever session is currently displayed.

Dive deeper

- Opens a submenu, or displays additional information. For example, in the Spanish Travel Phrases application, a right-tap on “Health and Medical” leads the user to a list of phrases related to these topics. For another example, in the World Series Champions application, a right-tap on the names of the teams that played in a given year reveals the MVP for that game.

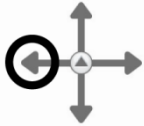
Confirm

- Confirm a command or respond to an onscreen question. For example, after the user chooses to delete a session in Paper Replay, the application shows

“Delete session?” on the display. To confirm the delete command, the user needs to right-tap.

Tap Left

tap left



A left-tap on the Nav Plus has different effects depending on the current application, mode, and state. In many ways, this is the opposite of the right-tap. A left-tap is used to exit, cancel, or rise up in the hierarchy.

Back to Parent Menu

- Left tapping on the Nav Plus slides the display text to the right to return to the parent menu. For example, left-tapping when in the Record Session submenu will navigate back to the Paper Replay application menu.

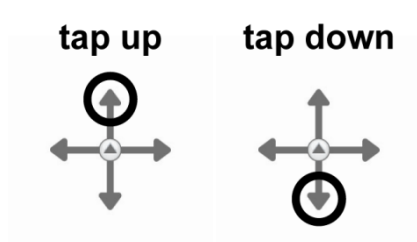
Exit

- If the application or one of its functions is running, left tapping out to the Main Menu exits the application or function. For example, left-tapping when in the Paper Replay application menu will navigate back to the Main Menu.

Cancel

- Cancel a command. Applications should consider left tapping as a “No” response from the user. For example, left tapping when asked “Delete this session?” for a Paper Replay session will not delete the session, and return to that entry in the “Delete Session” submenu.

Tap Up and Tap Down



Scroll Up/Down

- Up tapping and down tapping on the Nav Plus scrolls through a list of menu items that are at the same level, such as Record Session, Play Session, and Delete Session for Paper Replay.
- You can also use up and down tapping to scroll a list of items, such as a list of Paper Replay sessions that are available to play or delete. Menus and lists do not wrap. On reaching the top or bottom of a menu, the display shows a bracket in place of the scroll arrow. Continued attempts to move in that direction play a “can’t do that” audio instruction.

Application Names

In order to be seen and accessed within the smartpen’s menu system, applications must provide resources for the application name. There are two types of name resources: *text strings* and *audio files*.

Requirements

Application name as an English text string: Every application must at a minimum provide an English text string for the name, so it is accessible with the Main Menu when the smartpen is set to “English”. Including your application’s name in other Livescribe-supported languages (see the SDK for an up-to-date list) is optional. (Note: the English-language name will be displayed when the smartpen is set to any locale for which an application has not included a localized name.)

Recommendations

Audio of the application name being spoken: in addition to a text string for the name you’ve chosen, we recommend including audio of the name being spoken (to play when the name is displayed in the smartpen menu) to complete the user perception of your application’s identity. To include audio for an application name: create a smartpen-compatible sound file of the app name being read aloud. Next, enter the

file name in the “menu.txt” resource file, and the system will point to that file and play it as appropriate in the menu.

Application Menus

Standardized Menu Structure

The primary way users interact with a smartpen application is through its menu system. Whether simple or complex, the menu system must adhere to certain standards defined by Livescribe. A standardized approach to menu design will ensure a consistent user experience and make applications easier to learn and use.

When designing your application's menus, please observe the following rules:

Requirements

- Related items (such as top-level menu items, related commands, selectable options, and so on) must be organized vertically. In French Travel Phrases, for example, the main application menu consists of all the phrase categories (Transportation, Accommodation, etc.), and each submenu contains only related phrases.
- Actions on a specific menu item must be organized horizontally. In Paper Replay, for example, deleting a session is accomplished by right-tapping into the Delete Session submenu, right-tapping on a session to select it, and right-tapping again to confirm deletion.
- A left-tap must return the user to the previous item in the horizontal menu. Ultimately, the user can exit the application by tapping left enough times.
- A right-tap must move the user to the next item in the horizontal menu. Depending on the application, this may run a command, display information, or display sub-menu item.
- A down-tap must move the user to the next item in the vertical menu.
- An up-tap must return the user to the previous item in the vertical menu.
- The “top of menu” and “bottom of menu” bar appear when the user is viewing the first or last item in a menu. A right-arrow must appear on the display whenever an option is selectable or when other information is available via a right-tap.

User Experience Guidelines

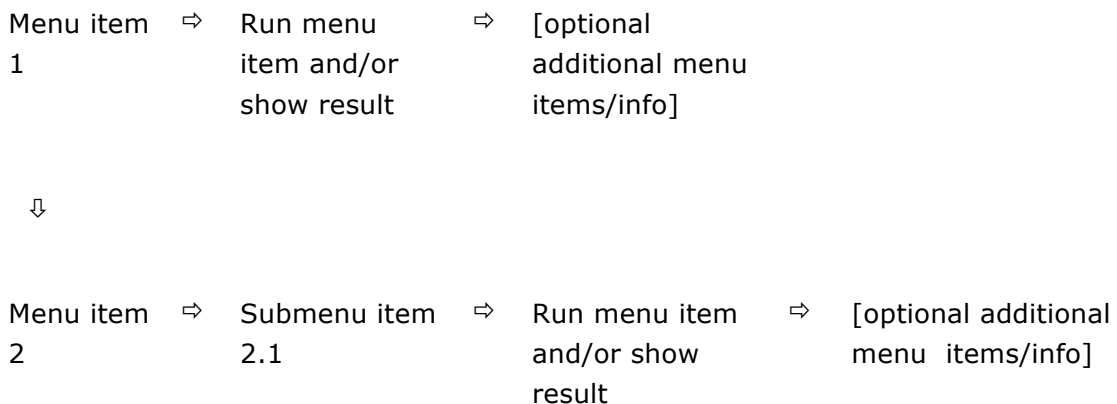
- All applications must have a Help or Instructions menu item that explains how to use the application or directs users to other resources, such as a URL to an online video or documentation.
- Applications must allow for the creation and use of Flick & Scrub regions whenever the current item on the screen is scrolling text (for horizontal F&S regions) or in a vertical scrolling list (for vertical F&S regions). See the Livescribe user manual for more information on Flick & Scrub.

Recommendations

- Recommended: Applications should have voice instructions that accompany the written text instructions.
- Livescribe applications use non-looping menus, and we recommend that you do the same. This allows Flick & Scrub to work well in menus, and also means that users can learn where an option appears within a menu and easily return to that spot.

The following diagram illustrates a standard menu structure for a smartpen application. Arrows in the diagram correspond to Nav Plus navigation by the user. For example, the ⇒ symbol indicates a right tap on a Nav Plus, and the ⇩ symbol indicates a down tap on a Nav Plus.

Application Menu



User Experience Guidelines



Submenu item 2.2 ⇒ Run menu item and/or show result ⇒ [optional additional menu items/info]

Menu item N ⇒ Run menu item and/or show result ⇒ [optional additional menu items or information]



Help ⇒ About ⇒ A few words or short sentence that describes the application



Version ⇒ The current version number of the application.



Contact ⇒ Developer’s customer service email address and/or website URL



User Experience Guidelines

⇒ [optional Help topic 1] ⇒ Help text regarding topic. ⇒ [optional additional Help text]



[optional Help topic 2] ⇒ Help text regarding topic. ⇒ [optional additional Help text]

Let's take as an example the menu structure of the Video Poker game. It is simple and follows Livescribe standards:

Video Poker

Start New Game ⇒ Starts a new game.



Instructions ⇒ Displays scrolling instructions



Hand Values ⇒ Displays poker hand 1 and values.



Displays poker hand 2 and values.



Displays poker hand N and values.

Menu Behavior

The default behavior for menus is non-looping (where reaching the end of the menu does not let the user jump back to the beginning). This will allow for proper functioning with the Flick & Scrub feature.

(An exception can be made for menus which the user scrolls through by tapping repeatedly on a button; for example, the “Instrument” menu in the Piano application.)

Menu Graphics

If a menu contains more than one item, the first and last items in the list should be marked with the “menu decorations.” These show a bar above the first item and a bar below the last item, and let the user know that they are at one end of the list and cannot scroll any further in that direction.

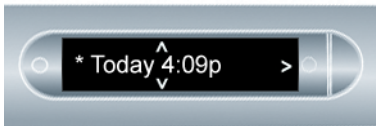
If a menu contains only one item, these bars should not be displayed.

For all intermediate entries in a menu (entries which have at least one entry above and one entry below), up and down arrow decorations must be displayed.

All entries which have right arrow tap actions must display a right arrow menu decoration.

Display

Display Area



The size of the Livescribe smartpen display is 96 pixels wide by 18 pixels high. Your application can write to the full display. However, the system may use some of the space to show system status. The space reserved for status is the System Tray. Its dimensions are 6 pixels wide x 18 pixels high, at the right edge of the display.

Elements that appear in the System Tray include:

- Low battery icon

- Low storage icon
- Background recording dot

If your application uses this right-most area of the screen, make sure that no critical information is displayed here. If any of the conditions that trigger the system tray become true, the tray will slide out and any appropriate icons will be displayed over the top of your application's display.

Horizontal Scroll Speed

If text is too long to be displayed all at once on the OLED, it scrolls to display the unseen characters. The speed at which text scrolls horizontally is a system setting. The user selects it to suit their needs or taste. Your application cannot modify the scroll speed. When designing applications, keep in mind that the user may have selected a speed slower or faster than the one you are using on your own smartpen.

The OLED Must Never Be Blank

A smartpen application must *never* display a blank screen. This is because a blank screen is the sign that a smartpen is turned off. Therefore, while the smartpen is on, the OLED must always display some pertinent information to the user about the currently running application.

For example, imagine that your application gives full audio instructions to the user for the current action, but you don't want to repeat the instructions onscreen. In that case, the OLED should display some brief but useful information. For instance, if your application is State Capitals and you provide an audio hint after 5 seconds of inactivity from the user, the OLED must at least display *Hint* or *Tip*.

Recommendation: Sometimes there will be a pause while the smartpen analyzes or processes something. This is often the case when an application has received handwriting input and is attempting to interpret it.

Make sure that your application is designed to leave something on the display while this processing is occurring. We recommend either:

- Leave the prompt on the screen until the result is available for display
- If a delay is known/expected, consider displaying a message such as "Loading..." or "Processing..." for the length of the delay.

Data on Display

Some applications write data to the display in response to a user request. Examples include the mathematical result provided by a calculator application or a definition provided by a dictionary application.

Applications that are not expecting input should exit if the user begins writing on Open Paper. However, a user will often want to write down the result from an application, and the Data on Display feature helps with this.

If your application provides information that you feel a user is likely to want to write down, and if that information is (or can be) too long to easily remember, then you should consider invoking Data on Display.

When the user begins writing, your application will still exit, but the information on the display will remain there for a while, to enable them to finish copying it down.

Providing Help

Livescribe applications communicate instructions to users as text, graphics and audio. You must *always* provide text or graphical instructions on the OLED display. Audio instruction may also be added, if you desire. Text/graphical instructions are mandatory because users may choose to mute their smartpen. All instructions need to be complete. Text and audio instructions do not need to be identical, but they each need to provide all the information a user needs to know to use the application or complete a specific command.

Help Menu Item

Livescribe highly requires that you provide a “Help” item in your main application menu. Your help should contain:

- About: A few words or short sentence that describes the application.
- Version: The current version number of the application.
- Contact: The developer’s customer service email address and/or website URL. Optionally, include the location of any additional help, such as an online instruction manual or instructional videos.
- Optional Help Topic(s) with help text about the topic. If it would help users to understand your application, you may want to include some basic

User Experience Guidelines

instructions, for example a sentence on the use of each of your application's major features.

Audio Punctuation Marks

The system also uses audio punctuation marks (APMs) to convey success or failure of a command. APMs also indicate transitions from one state to another or from one menu to another. APMs are usually short beeps or chimes. You can use APMs in your application as well.

State	Description	File
Execute/Select/Confirm	User is choosing/selecting/activating current item	SL_Select
Non-final step in a sequence	Confirmation of a non-final step in a sequence	SL_Ack
Exit/Back Up/Cancel	User is exiting from current list/navigating up a level/canceling current option or item	SL_Exit
Move Up/Down in Current List	User is moving through current list of items at same level	SL_NavDown
Action Not Available/End of List	User is touching active item which has functionality which is not available now, including trying to move up or down when at the top or bottom of a list	SL_NA
General Active "Click"	For use for items which are active and the above states don't fit, or for volume buttons	SL_Tap

Application Deactivation

Once started, Livescribe applications continue to run. Instead of terminating, applications change from an active state to an inactive state. For example, when a user changes from one application to another, the first application is deactivated and the second one is activated if it is already running, or started if not. Application switching is the most common cause for an application to deactivate. However, there are other cases as well. When designing your application, you should consider these cases and program the proper response:

Application switching	The current application deactivates because the system switched to another application.
Notes Mode	The current application deactivates because the user switched to Notes Mode. Notes Mode is activated when users write on Open Paper and the current application is not expecting any Open Paper input.
Self deactivation	The current application deactivates because the application changed its own state.
System shutdown	The current application deactivates because the system is shutting down.
System event	The current application deactivates because of a system event.

One-Second Deactivation

Due to the nature of Livescribe smartpen computing, applications must respond to user input immediately. When a user switches to another application, the current application must deactivate in **one second or less**. Unlike a PC application, which normally asks tidying-up questions such as whether files or settings should be saved,

a smartpen application must not ask for any input from the user before exiting. If your application needs to save data or ask the user questions, make sure it does this along the way, rather than waiting until a deactivation takes place.

Livescribe Standard Controls

The Livescribe Standard Controls, such as the Paper Replay controls or the Settings controls, can be included in a Fixed-Print application. For information on how to include them in your application, see "Livescribe Standard Controls" in *Developing Paper Products*. The Platform SDK includes the graphics for these controls, so you can make them part of your Fixed-Print paper product. Please note that when a user taps on these controls, the standard Livescribe applications or system functions will be triggered and/or launched.

Do Not Repurpose Standard Controls

Smartpen applications must not, however, use standard control graphics and assign them different functionality. For instance, a Nav Plus graphic must always behave the way users expect it to behave. In short, smartpen applications must not repurpose the existing Livescribe Standard Controls.

Open Paper

Termination Events

Termination Events (TEs) are notifications from users that they are finished responding. Whenever a user is prompted for input by a smartpen application, a TE is how the user signifies that they are through writing or drawing their input and are ready to proceed. The most common forms of TEs are:

- Timeout
- Double-tap on Open Paper
- Special characters

In most cases, the timeout and double-tap TEs should both be accepted.

There are some cases where a timeout TE could cause a user's input to be taken before they are finished writing. Examples are where a user might need to refer to other materials or think carefully about their input, such as for math calculations or

when writing a word in a foreign language. In these cases, the timeout TE should not be included.

Timeout

Timeout assumes the user is finished responding when there is no smartpen activity on paper for a certain timeout period. The recommended default timeout is one second. However, some applications might want a longer timeout setting.

Example: In the Piano application, the user is prompted with "Write the letter 'I'." This is your instrument icon." Once the user begins writing on OP, a pause of one second constitutes a TE, and the application then moves on to the next request. Whatever the user has drawn before the timeout will be used as the instrument icon going forward.

There are times when a timeout TE are not appropriate. Think about your user's state of mind and the typical activities they may be engaged in while using your application or function. For example, in QuickCalc, when the user is entering a math problem, the timeout TE is not used. This is because it's easy to imagine that a user might need to pause to find the numbers they want to use in their mathematical expression. If a user is adding up receipts, they might even be setting down the pen between each number. So in this case, other TEs are accepted for finalizing the entry of the expression, but a timeout is not used.

Double-Tap on Open Paper

Another way for users to indicate they have finished responding is for them to double-tap on Open Paper. You can design your application to instruct users to double-tap somewhere on Open Paper when they have finished writing.

This is the most common of the TEs, and it should be included in all cases where a TE is used. Users will get used to using double-tap to finish their entry and move quickly on to the next step.

Special Characters

You can also have the application look for special characters that indicate the user has finished entering data. For example, the Calc Quick Command application assumes the user has finished entering numbers when the user writes "=" (the equals sign). Use special character TEs only where users could easily be confused were the option not available.

Special characters are specific to an application or situation, and are included where they align with how a user might reasonably expect an application to work.

Prompt and Believe

For some applications, Livescribe uses a protocol called Prompt and Believe. This protocol is for an application that prompts a user for a specific input but does not analyze or confirm the actual input. Instead, the application assumes (or believes) that the user entered the proper value.

For example, an application might prompt a user to create a circle and put a letter “r” in it to create a “Reminder” button. Instead of using character recognition to determine that users actually enter an “r”, the application trusts that they always do. The application simply notes the area the user wrote in and assigns the entire bounding area of their strokes to the “Reminder” function. The user can write a circled “r” as directed, or could write the word “Reminder” or draw a picture of a finger with a string tied around it. From then on, the application treats whatever the user drew as the button and invokes the proper code when the user subsequently taps on it.

Prompt and Believe allows for flexibility and creativity for the user. However, we’ve found that it helps when the instructions suggest something for the user to draw. When the directions are completely open (e.g., “Draw a control”), sometimes users worry about getting it wrong. It’s similar to the old “press any key” instruction on desktop computers, which had to be replaced because nervous users sat staring at their keyboards, looking for a key labeled “Any.” We recommend language such as “Draw your ‘Reminder’ button, and then double tap.” This tells users what the button will be used for, and also gives some suggestion as to how to label it.

Dynamic Active Regions

Dynamic Active regions are user-created areas on open paper pages that are saved by the application and which have ongoing functions. For example, an application may ask a user to create a button by drawing or writing a word. The user draws the button, and the application saves the bounding rectangle as a dynamic active region. When the user later taps in that region, the specific function associated with that region is triggered.

Another example of dynamic active regions is when a user writes something as input for an application. This could be a word to be translated, a mathematical expression for calculation, or a number written as a variable or setting (such as how many dice to roll).

Wherever it makes sense and could be desirable to the user, the application should ensure these regions remain active, even after the application is deactivated or destroyed. Keeping these regions active ensures that tapping on them later triggers the appropriate function in the application. For example, an application that does translation or calculation, should repeat the same function performed when the information was first written. This allows users to quickly get back to look-ups or calculations they already performed. The application should save the regions across application launches, so that any time a user returns to them, they retain their function and value.

Shaping Dynamic Active Regions

In most cases, dynamic active regions created on Open Paper should correspond to the area written on by the user. It is usually acceptable to create the active region as a rectangular bounding area around what is written/drawn by the user. In some cases (such as when the user draws a large diagonal line) the shaping should employ polygons or some other way to create regions more closely corresponding to exactly what is drawn. The goal is for ease of use, so that users can clearly see which areas of their page are mapped to applications or functions. In most cases, the application should not create dynamically active regions on Open Paper where the user has not written or drawn.

Activating Functions

When a user taps on a dynamic active region, the application that owns that region will launch. This is the default behavior of using dynamic active regions on Open Paper.

In general, if a user has drawn a dynamic region, the user's expectation is that when tapping that region at a later time will launch the application and perform the function associated with that region. For example, if a user taps on the "Deal" button of Video Poker, the application will launch and then deal a new hand.

In some cases, the dynamic region is associated with a function that is not actionable at the time. For example, when Video Poker is not active, and a user taps on a pre-drawn "hold" button, the application launches. However, there is no active hand, and therefore, nothing to hold. In this case, the hold function is not actionable. Video Poker handles this situation as follows. Your penlets should follow a similar approach:

- Display a helpful message, such as "First tap the DEAL button to deal a hand"
- Play the Action not available audio punctuation mark

- Position the application in the main application menu.

Paper Replay

Background Recording

Paper Replay can record in background mode. This allows users to start recording a Paper Replay session and then start or switch to another application. When they do this, recording continues. The Livescribe smartpen display shows a flashing dot on the System Tray to indicate recording is going on in the background. Any application that is active cannot record or play audio when Paper Replay is in background recording mode. If your application requires audio recording or playback, you should test if the Livescribe smartpen is already in background recording mode and if so, instruct your users to turn it off before running your application. For an example of this, try the Piano application while recording. Livescribe instructs you to turn off recording before running the Piano because the piano application is an audio-centric application.

Snapback

When in background recording mode, Paper Replay can become the foreground application in several ways. When this happens, the Livescribe smartpen system will deactivate the application that was in the foreground. The most obvious way is in response to action from a user. Here are some actions that can restore Paper Replay to the foreground:

- Tapping on any Fixed Print Paper Replay control. Even though some controls have no effect during recording, tapping on them brings Paper Replay to the foreground.
- Tapping on a Fixed Print area that is associated with Paper Replay but not assigned to a particular button will bring Paper Replay back to the foreground.
- Doing any other action that would normally start Paper Replay, such as tapping on a session, or launching Paper Replay from the Nav Plus.
- Writing on Open Paper while the current foreground application is not expecting Open Paper input. Examples are Fixed Print Calculator at any time, since it never takes Open Paper input, or Piano after the user has created keyboard and buttons because it is not expecting any Open Paper input at that point.

User Experience Guidelines

Paper Replay in background recording mode automatically returns to the foreground if the current application does not receive input from the user for five seconds. This feature is *Snapback*. For example, if the Calculator application is the current foreground application and the user pauses for five seconds without providing input, Paper Replay snaps back to be the foreground application. Snapback will deactivate the application that was in the foreground.

While Paper Replay is recording in the background, it does not receive any new strokes made by the user, as those belong to the current foreground application. Snapback is implemented to protect the user from accidentally remaining in background recording when they believe that Paper Replay is active in the foreground and is capturing and linking their strokes to their audio.