|  |  |  |  |
| --- | --- | --- | --- |
| **Subject**: | JAWS Split braille display Feature | **Date**: | 23/06/2023 |
| **Version**: | 1.0 | **From**: | Joseph Stephen |
|  |  |  |  |
|  |  |  |  |

# Summary

JAWS 2024 introduces a new revolutionary Split Braille feature. This feature will expand the ways in which braille is presented. We can now show braille output from two different locations simultaneously on a single line Braille display. We can offer a variety of views for line mode consisting of expanded capabilities which are not currently available to braille users. We can create and package choices for structured mode braille, which give the user greater flexibility for braille presentation. We can also provide the user with the ability to create multiple structured braille views suitable to very specific scenarios and choose which views to use for those scenarios. And we can add braille translation comparison views which can be used either for training or for braille translation table comparison.

A new SetBrailleView dialog has been implemented, which offers the user application and even context-specific options for Split Braille views.

This dialog may be invoked from the QWERTY keyboard using Alt + JAWS key + V, or the Braille display hotkey Braille Shift1+Dots 127 (b with dot 7 for Braille view on a Focus display). Third party vendors may optionally add a keystroke to invoke this dialog. See the keymap section below for details.

This dialog consists among other things of a list of applicable Split views, a button to swap the sides of the display and a description field to describe each view.

These views include the following:

## Buffered Text.

This view allows the user to read the focused control or document in region one of the display and text from a buffered document in region 2. The user can pan through both regions independently and copy text from the buffered document to the live document easily.

### Use Cases.

1. Buffered text is available for easy review and reference, rather than needing to remember it.
2. It is useful for the comparison of information between two different sources. (Most Braille readers can read independently with both hands).
3. Keeping text available for researching a topic.
4. Keeping text of a question or math equation in view while composing an answer without losing reference to both point of review and point of composition/editing.
5. Keeping a long string of text or numbers not easily remembered for temporary reference.
6. The buffered text can be copied and pasted into a document or edit field on a web form where desired.

## Annotations view

In Annotations view, the first region shall show text from a document while the second region shall show annotations from the first region if the cursor is within a reference mark or text to which an annotation applies. (If there is no annotation at the cursor, the full display will be used for the text of the line).

Panning of each region shall be independent if an annotation is being displayed.

### Use cases

Text from annotations such as comments, revisions, footnotes and endnotes can be shown at the same time as the text to which they refer. The user can edit and read the document text while keeping the annotation text in view.

## Attributes view

In attributes view, the first region shall show text from the document while the second region shall show attribute indicators corresponding to the text in the first region.

If a character has multiple text attributes, the attribute indicator shall be rotated as in the non-split attribute rotation mode (the rate of which is configurable in Settings Center).

Panning shall be synchronized.

Pressing a routing button over an attribute indicator will be the same as pressing a router over its equivalent character, i.e. the caret will move to the corresponding character in the document.

### Use case

Braille can now show specifically whether text is underlined, bolded, italicized and so forth while the text is shown. This is most useful for proofreading and editing. The user can literally read the text with one finger and follow with a second finger in the equivalent attribute indicators.

## Speech History

In speech history view, the first region shall show the normal Braille information while the second region shall show text from the speech history buffer.

The user shall be able to pan through the speech history independently to the main Braille information.

### Use case

This mode is most useful when running without speech output, to enable review of speech history which is not available without switching to the speech output mode or invoking and dismissing the Speech History feature. Necessarily, speech output includes more information than Braille so augmenting the Braille with the speech equivalent is often useful.

## JAWS Cursor Split

Typically the user would route the JAWS cursor to the PC cursor and then locate a window or text of interest and park it there, then switch back to another cursor such as the PC cursor and activate this mode.

In this view, one region shall show the PC or other cursor location, while the other shows the line of text at the JAWS cursor location.

Panning shall be independent.

### Use cases

Percentage data or progress bar may be monitored for a long-running task, even when the percentage data or progress bar is non-focusable.

Application status bar may be monitored.

Document title bar may be read in a multi document interface application.

## Window Text

This mode monitors the window of the active cursor at the time the feature was invoked (or could be invoked from scripts given a window handle). Once enabled, the user can set focus elsewhere while keeping text from the window available in region 2 of the display. The difference between this and buffered mode is that if the text changes in the window, it is updated instantly on the display. Also, in this mode, all the visible text from the window is available for reading, unlike the JAWS cursor split which only makes the text on the line containing the JAWS cursor available.

The same ability to copy and paste from the window text to the live Braille line is available in this mode.

### Use Case

This view is specifically designed for scripting custom applications where monitoring a window in the same application makes work more efficient.

## Translation Split

This view shall display two different translations of the same braille--one in each region: e.g. US Grade 2 English in region 1 and Grade 1 in region 2.

Panning shall be independent as one region will likely be different in length due to the differences in translation.

### Use Cases

1. Assist in learning correct braille patterns for different translation tables. For example, show English computer Braille on line 1 and German computer Braille on line 2.
2. Clarify any ambiguities due to specific braille translation tables. For example: Show Grade 2 English on line 1 and Grade 1 or Unified English Braille on line 2.

## Scripted Application Views

Other Split Views shall be included for specific applications. For example:

### Outlook

The user shall be able to navigate their inbox and read the full message in region two of the display. They’ll be able to pan through the text of the message without leaving the inbox.

### Powerpoint

The user shall be able to review a slide show presentation in one region and the speakers notes in the second region, independently panning through each region.

### New Applications

Scripters shall be able to add custom split views to any application with many new built-in functions in the JAWS scripting language. Documentation shall be made available to assist in this task.

## Key Maps

Third-party Braille vendors must choose key bindings for the following scripts in order to take advantage of this new Split functionality. They are listed in order of importance.

### BrailleSplitPanRight

### BrailleSplitPanLeft

### BrailleSplitNextLine

### BrailleSplitPriorLine

These four key bindings are essential to the Split feature and should be mapped to appropriate key bindings by 3rd party vendors. They allow the user to independently pan and navigate through the Braille in the split region of the display. Without these keys, the new feature is unusable on the display.

For Focus displays, these have been scripted into the right front rocker for BrailleSplitNext/PriorLine and the right top rocker for BrailleSplitPanLeft/Right if a split mode is active.

### BrailleSelectText

Many vendors may already support this. The Focus display uses

**Braille Shift+Routing=BrailleSelectText**

This not only allows text to be selected from the Braille display, in Buffered Text Split view, it will automatically allow selecting and copying of buffered text to the clipboard for pasting into the active application.

### toggleBufferedTextMode

This toggle allows the user to quickly capture text from the current document and enable Buffered Text view from the display. (It is available from the SetBrailleView dialog, but having it on a Display key binding makes it much faster to capture text or refresh text in the buffer without having to invoke the dialog.)

This toggle has been added to the Focus display keymap as

**Braille Shift1+Dots 2 3=toggleBufferedTextMode**

We suggest 3rd party Braille vendors add a key binding to this script to facilitate the ease of access to this new vital feature.

### SetBrailleView

alt+JAWSKey+v=SetBrailleView

While a QWERTY key binding has been added to invoke the SetBrailleView dialog, we suggest 3rd party vendors add a Display specific key binding to invoke this script from their Braille display. For example, the Focus display uses

**Braille Shift1+Dots 1 2 7=SetBrailleView**