

Braille2000, LLC

6801 Southfork Cir, Lincoln, NE 68516
Phone: (402) 423-4782, Fax: (402) 423-5154
<http://www.braille2000.com>

May 2017
(Version 2.268)

A Guide to *Braille2000 V2*

Overview

Braille2000 Version 2 (V2) is a more powerful Braille2000 that replaces the version1 (V1). This document presents the major extensions and differences and also describes in detail the entire user interface for V2.

Braille2000 V2 is compatible with Windows 8 and Windows 10.

Braille2000 V1 and V2 can operate at the same time, but not using the same Internet license.

Braille2000 V2 includes support for UEB.

October 2016 status

The current work area is math braille, which offers added difficulty because the entire concept of "math braille" is presently undefined, given the verbal commitment "Keeping Nemeth" with the actual commitment called "Provisional Guidance" (which is only Nemeth-like). It will require about triple the regular work to provide support for those two codes plus support for UEB math (which the National Library Service wants to pursue).

At this time, V2 is still a "beta" product: some behaviors are incomplete and a few design features are missing entirely, yet it can still do a lot. Technical support is available should you have problems.

This "beta" release has undergone modest testing. It is designed to open V1 (.abt) files for further editing and compatibility evaluations show this is usually the case (please report any problems). The following "temperamental" aspects are being studied:

- Undo (in some situations)
- Vertically sectioned tables
- Spell checking (no suggestion box yet)
- Speedbraille™ definitions (a few dialog boxes do not yet generate transactions)

Major Differences

1. Different filename suffix (.b2k)
2. Up/Down-arrow into yellow and gray
3. Different Speedbraille™ macros
4. No aqua-tint to the Print View (aqua-tint now indicates “block” material (lists and tables))
5. Different procedure to import RTF file, revised/extended translation typing directives
6. Red triangles are now circles and can be clicked to highlight whole paragraphs
7. Braille Page Break is now handled by Page Control
8. Support for adjusted margins (left and right) in Layout Control
9. Dropping of support for V2 on Windows95, Windows98, Windows-ME
10. There are two cursor positions between adjacent cells

Major New Features

1. UEB
2. Additional views
3. Table editing/formatting
4. Tab stops in prose
5. Additional keyboard behaviors
6. Multi-select
7. Blue blocks (adjacent paragraphs with a special function or behavior, including tables, nested lists (2011 Formats), notes (e.g., footnotes, future enhancement), auto-generated tables of contents (future enhancement), drawings and drawing spaces).
8. Math (not complete, but its getting there, for UEB/Nemeth/Provisional Guidance)
9. Dynamic boxing lines and creation tools

Installing Braille2000 V2

You can get the setup file BRL2000B_Setup.Exe from www.braille2000.com (click the V2beta link and then click the Download button). If your browser offers to let you "Run" the download, do that. Else save the file to a convenient location (the Desktop is handy) and after the download is complete, launch the file to run it. You may need to turn off your anti-virus system during the installation. You **will** need to have administrator capabilities (use a Windows logon that is at the Administrator level).

V2 will honor your license for V1 **if** your license supports updates (“Premium” and “Subscription” types of licenses support updates while “Budget” licenses support updates only during the first 90 days after purchase). If your V1 license does not support updates, you will have to use “Evaluate Only” with V2 and it will run in “demonstrator” mode that does not support production work (it overwrites your file with “demo”). **Notice: do not force a demo version to save your V1 (valuable, preexisting) work: it will be corrupted with “demo”.** If you have a Budget license, you can upgrade for a fee.

User Interface

Display

There are four display modes: A (Ascii), B (braille, i.e., simbraille), P (print), and S (source). These can also be called the Ascii View, Braille View, Print View, and Source View, respectively. The mode can be selected via four adjacent buttons in the toolbar. The Ascii and Braille views present the braille page, using Roman characters (so called ASCII-braille), or simbraille, respectively. The Print and Source views present print text. The Print view retains the same size and shape as the braille page, with print text positioned to match its braille equivalent. When the print equivalent and/or auxiliary material (such as an embedded image from a NIMAS file) would cause the layout to deviate from that of the braille page, the auxiliary material is omitted and/or the tall math expression is replaced with its equivalent in simbraille. The Source view allows the page layout to deviate (usually in the vertical direction) from the braille page, thereby allowing tall math expressions and auxiliary material (such as images) to show in place. The Source view thereby gives a better print depiction of the material while the Print view allows the viewer to see the layout of the resulting braille page.



The default background color scheme is as follows:

Color	Usage
White	normal prose
Lt. Blue	special text blocks (e.g., table, list, etc.)
Yellow	generated page data (running head, page numbers etc.)
Gray	fill (empty voids as well as automatic format-required punctuation)
Red	warnings (spelling-suspect words)

Note that the light cyan tint of the V1 Print View is no longer the default background color for normal prose in the V2 Print View (white is used in all viewing modes for normal prose).

The first line of each paragraph is marked with a colored circle in the left border of the editing panel. The color changes to blue when the line is also the first line in a block (table, nested list, note, drawing). The interior of the circle becomes colored when the entire paragraph is selected (e.g., highlighted). Clicking the circle with the mouse selects the entire paragraph. Blocks are shown with a powder-blue background tint (to remind the user that editing behaviors are specialized in such areas of the file, e.g., the use of the Tab key within a table).

Keyboard and Cursor

Navigation

In V2, the blinking cursor can exist within yellow and gray areas. As before you cannot directly enter keyboard data into such regions.

The up- and down-arrow keys (and the mouse) can move the cursor to any line regardless of color, however the left- and right-arrow keys can move the cursor only to editable positions, i.e., not into yellow or gray. If the cursor is already within yellow or gray (via the up- or down-arrow or the mouse), either left/right arrow key moves to beyond the yellow or gray.

By default, the leading and trailing blanks on each line are jumped over when using left- or right-arrow. This is controlled by “Arrow keys skip margins” in the dialog box reached via Adjust and Keyboard. Regardless of that setting, you can arrow-key into a margin using Alt-left-arrow or Alt-right-arrow.

Document control points (annotations) may exist between cells (between characters in the print view) and can be visualized by small yellow triangles (use Adjust / Display to control whether or not they show). The left- and right-arrow keys move the cursor cell by cell. But if you look carefully, you will see that there are actually two possible cursor positions between the cells, one is a hair to the left of any annotation and the other is a hair to the right of any annotation. The left- and right-arrow keys do not provide for any hair-width cursor movements (you can do that via the menu using Menu / Edit / Insert / Flip gap bias to move the cursor from one side of annotations to the other, at the same cell gap). But you can indirectly cause that effect using left- and right-arrow keys because left-to-right (cell-to-cell) movement by the right-arrow key lands the cursor on the left side of any annotations, while right-to-left (cell-to-cell) movement by the left-arrow key lands the cursor on the right side of any annotations. Thus right- and then left-arrow places the cursor to the right of annotations while left- and then right-arrow places the cursor to the left of annotations.

The two inter-cell (gap) cursor positions can be distinguished visually. Here is an enlargement of the cursor positioned *before* vs. *after* the annotations. Keyboard entry and Paste functions occur *after* annotations when the cursor is to the right, but *before* them when the cursor is to the left. If the annotation is, say, a bookmark, this positioning determines whether new material is before vs. after the bookmark. If the annotation is, say, a Code change, this positioning determines whether the new material is in the first code versus the second code.



When used with the Ctrl key the arrow keys behave as follows:

Key	Effect
Ctrl-left-arrow	jumps left to the next word beginning
Ctrl-right-arrow	jumps right to the next work ending
Ctrl-up-arrow	moves up one line and scrolls down one line*
Ctrl-down-arrow	moves down one line and scrolls up one line*

*Page View Scrolling is turned “off”

The Home and End keys jump the cursor to useful locations, as per the following table:

Key	Effect
Home	first non-blank position in the same line
Home x2	first position in the same line
Home x3	first non-blank position in the same paragraph
Home x4	first non-blank position in the same page
Home x5	first non-blank position in the file
Ctrl-Home	first non-blank position in the file
End	after last non-blank position in the same line
End x2	after last position in the same line
End x3	after last non-blank position in the same paragraph
End x4	after last non-blank position in the same page
End x5	after last non-blank position in the file
Ctrl-End	after last non-blank position in the file

The notation xN (as in x2) denotes the number of times the key is pressed in succession without any other keypress intervening. Thus if you press Home three times in a row the cursor jumps to the start of the paragraph.

The Pg Up and Pg Down keys move the cursor up or down by the number of lines showing on the screen (usually 25). When Page View Scrolling is “on” and the screen shows at least the number of lines as on a page, the cursor moves exactly one page regardless of the number of lines actually showing. When Pg Up or Pg Down is pressed with Ctrl then Page View Scrolling is turned “on” (unless the screen shows fewer lines than one page) and the cursor moves to the first non-blank position on the page (as if Home x4 were implied).

Page-View Scrolling

There are two scrolling modes: line scrolling and page scrolling (called Page-View). In page scrolling the display scrolls by whole pages, but this mode is available only when the editing panel shows at least a whole page at a time. The current page scrolling mode is indicated by the appearance of the “pv” button in the extreme lower-right corner of the window (where the vertical and horizontal scroll bars meet). The legend (“pv”) is dark when page scrolling is active and grayed out when line scrolling is active. Assuming the screen shows a whole page, the status can be toggled by clicking the “pv” button with the mouse. The setting is also changed by using Ctrl-Up/Down-Arrow and Ctrl-Pg-Up/Down.

Editing

The following keys are used to change the content of the document:

Key	Effect
symbol keys	<p>In typewriter input mode, the symbol keys insert or overtype the position where the cursor is located. If the cursor is within a yellow or gray area, it first jumps out of yellow and/or gray moving the same direction (forward or reverse) that it moved when entering the yellow or gray area (mouse click cursor positioning implies forward momentum).</p> <p>In braillewriter input mode, the keys sdfjkl form braille cells as if a braille typewriter. Other letters trigger SpeedBraille™ Keys. The digits 0 to 9 (shifted or unshifted) input the corresponding ASCII-braille cell unless a SpeedBraille definition is given for that key.</p>
Enter	<p>When the cursor is not within a table, the enter key inserts a paragraph break. Within a table, the enter key moves the cursor to the first column of the next row or if the cursor is in the last row, a new row is created and the cursor jumps to the first column of that row.</p>

Backspace	<p>Without the Ctrl key, the backspace key moves the cursor like left-arrow and then deletes one character or cell (as per Delete, below) except that when moving across a non-deleteable boundary (a column separation in a table) no delete action is performed.</p> <p>With the Ctrl key, the backspace key merges the current paragraph with the preceding one, leaving one space between the two.</p>
Delete	<p>Without the Ctrl key, the delete key deletes the character to the right of the cursor without moving. This applies also to delete-able boundaries (paragraph boundaries and tabs within tabbed text). If the cursor is to the immediate left of a non-deleteable boundary (a column separation in a table), the delete key merely moves the cursor as per right-arrow.</p> <p>With the Ctrl key, the delete key merges the current paragraph with the following one, leaving one space between the two.</p>
Tab	<p>When the cursor is not within a table, the tab key inserts a tab stop. Within a table, the tab key jumps to the next table column or if the cursor is in the last column, a new table row is created and the cursor jumps to the first column in the new row. Within a table, Shift-Tab moves the cursor to the previous column and if the cursor is already in column 1 it moves the cursor to the last column of the previous row (if already in the first row, the cursor remains in the first row).</p> <p>When the cursor is not within a table, Ctrl-Tab increases the style level (e.g., increases indent level) and Shift-Ctrl-Tab decreases style level.</p>

Highlighting

In Version 2, a highlight is one kind of *Selection*. A selection is a dynamic indicator of one or more selected phrases in the document, such that you can see and/or manipulate those selected phrases all at the same time. The “highlighting” is merely the selection you create via the techniques listed below. Elements highlighted can be transferred to a named selection and stored with the file for later use. A selection can also be established via “Find All” and via document structure patterns, such as by using the NIMAS "dtbook" panel. A file can have zero or more named selections.

Unlike Version 1, selections (including highlighting) cover only prose and not margins (implied blanks usually at the extreme right or left of a line) nor generated text (yellow material) nor void fill (gray material). This difference has a few subtle effects that are different, such as the implied

behavior when replacing a selected (e.g., highlighted) phrase that spans the indent or runover margin area of a paragraph (such areas, containing virtual blank cells, could be highlighted in V1 but not in V2, with the corresponding ability for V1 to make changes to indent or runover settings (not always intended by the user) when manipulating such a selection).

Named and stored selections adjust dynamically to editing and can propagate through cut/copy/paste operations. Named and stored selections can have a “Post-it” note attached to each selected phrase, for example, to convey proofreader comments back to the transcriber.

The “Information Card” concept from Version 1 has been retained—it is a whole-document note whereas the named selection can give a note per selected phrase.

How to highlight... **By using the mouse**

Regular linear highlight

Text can be highlighted by dragging the mouse across the region to be highlighted (position the mouse at the start of the highlight, press and hold the left mouse button while moving the mouse to the end of the highlight, release the mouse button). During mouse action, the start location is fixed and does not move but the end location follows the mouse so long as the left mouse button remains pressed. After the highlight is defined, both ends of the highlight can be adjusted further by holding down the Shift key while you re-drag the mouse to re-establish the start or end location.

Multi-highlight

If you want to create a multi-highlight (to highlight multiple non-contiguous phrases), first form the an initial highlight as described above. Highlight a subsequent non-contiguous phrase by holding down the Ctrl key as you start dragging the mouse for the next phrase. Once the highlighting for the new phrase is established, you can use all of the techniques given above to adjust the end-points of the most recently created highlighted phrase. This entire process can be repeated to add additional non-contiguous phrases to the highlighting.

Block highlight

You can create a block highlight (only in the Braille View or ASCII View) by holding down the Alt key as you begin dragging the mouse. When you start a block highlight, any previous highlighting is cleared: you cannot form a multi-block highlight.

By using the keyboard

Regular linear highlight

Text can be highlighted by holding down the Shift key while using any of the cursor positioning keys: the material from the original cursor position to the final position will become highlighted. To discard the highlighting, press Esc.

If “Persistent highlights” (see below) is set to “off”, while the cursor remains at the end of a highlighted phrase you can hold down the Shift key and re-adjust the highlight using arrow keys.

If “Persistent highlights” is set to “on”, you can use arrow keys to move the cursor to either end of the highlight and then hold down Shift and re-adjust that end by using arrow keys.

Multi-highlight

To create a multi-highlight using the keyboard, “Persistent highlights” (in Keyboard options, reached via Adjust and Keyboard) must be set to “on”. In that setting, the arrow keys (by themselves) do not clear the highlight (you can clear the highlight using the ESC key). Highlight the first phrase using the Shift key as described above. Then move the cursor to a new location using just the arrow keys. Then create another highlighted phrase again using the Shift key. If the cursor is right at the end-point of the last-highlighted phrase, using the Shift key with arrow keys adjusts that end-point rather than begin a new highlighted phrase.

Block highlight

To form a block highlight using the keyboard, hold down the Shift and Alt keys while you use arrow keys to expand the size of the block. Begin using the left or right arrow keys to establish an initial breadth to the block highlight (otherwise nothing shows on the screen). Only arrow keys work to form a block highlight (not PgUp or PgDn that work to form a linear highlight).

Note that the block highlight is a braille concept (a block of braille cells with straight sides), and can only be created in the Braille View or the ASCII View.

Via mouse clicks

You can highlight units of prose via multiple left-clicks with the mouse:

Number of clicks	Effect
one	The cursor moves to the mouse pointer
two	The word at the mouse pointer is highlighted
three	The line at the mouse pointer is highlighted
four	The paragraph at the mouse pointer is highlighted
five	The page at the mouse pointer is highlighted
six	The entire file is highlighted

Multiple clicks must occur with no more than a half-second between clicks (i.e., at a moderate pace). If the Ctrl key is pressed during the mouse clicks, the new area is added to the current highlight as a separate phrase (else the current highlighting, if any, is removed and the new area becomes the only area highlighted).

You can also click in the screen region immediately to the left of cell 1, where you see paragraph circles and optional page-line numbers. Clicking on a paragraph circle highlights the whole paragraph. Clicking on a line number or nearby but not on a paragraph circle highlights just that line. Having begun by clicking a line number or a paragraph circle, you can drag the mouse up or down to include more lines/paragraphs.

Changing emphasis...

After highlighting some text (braille or print view, single- or multi-highlight), you can click any of the four emphasis buttons (each a toggle) to adjust the emphasis (typeform) of the highlighted material. The buttons are for Bold, Italics, Script, and Underlined.



Additional emphasis (typeform) control is available via the menu at (Menu / Adjust / Emphasis).

Cut / Copy / Paste

Selected text can be cut or copied to the Clipboard. Both operations put a copy of the selected text on the Clipboard. Only the cut operation then deletes the selected text from the document.

When the selection has multiple phrases, they all arrive at the Clipboard as separate paragraphs. By using special cut or special copy (via right-click to the corresponding button) you can target the cut or copied text to a new workarea rather than to the Clipboard itself.

Text is managed by various kinds of “marks” such as “paragraph break” and sometimes “line break”. This discussion is about paragraph marks.

When a phrase is copied to the Clipboard, it has optional beginning and ending (“exterior”) marks (invisible to the user) that will be written here as <p>. A phrase can also have interior marks. It is the exterior marks that determine how the phrase joins with other text when pasted. A copied phrase can thus be represented as either of the following. (Interior marks are not mentioned, as if none are present.)

Abstract notation	Behavior
phrase	Paste without adding paragraph breaks
<p>phrase<p>	Paste with leading/following paragraph breaks

The exterior <p> mark is part of the information copied to the Clipboard. **It is indicated or not by the manner of highlighting:** when one or more whole “mark units” (e.g., paragraphs) are highlighted, exterior marks are included; when other than whole units are highlighted, exterior marks are omitted. Exterior marks for paragraphs can also be added by using special paste. When a whole paragraph is highlighted, the paragraph’s “paragraph circle” (the beginning of paragraph visual on-screen indicator) becomes colored-in. The marks being discussed here manage the creation of new paragraphs during paste, as follows:

Cursor location for Paste	Effect (based on N_x/N_i numbers of exterior/interior <p> boundary marks pasted)
At the start of a target paragraph	Creates $N_i + N_x/2$ number of new paragraphs
In the middle of a target paragraph	Creates $N_i + N_x$ number of new paragraphs
At the end of a target paragraph	Creates $N_i + N_x/2$ number of new paragraphs

Note: in this scheme leading/trailing <p> marks always go together (both or none), i.e., N_x is always 0 or 2.

Following a Paste operating, the cursor is situated at the end of the added material. This behavior can be changed: right-click the Paste button and select "Cursor afterwards" and then choose the behavior you want, it can be "at End" (the default), "at Beginning", or "Highlight" (the pasted material becomes highlighted).

Keyboard actions (shortcuts)

The Ctrl key used with certain letters will trigger standard menu actions, as follows:

** denotes actions unique to Braille2000 (otherwise the action is common throughout Windows)

- Ctrl-a Select (highlight) all text
- Ctrl-b Bold (toggle)
- Ctrl-c Copy (ordinary copy, not for copy to new workarea)

Ctrl-d	
Ctrl-e**	Emboss
Ctrl-f	Find
Ctrl-g**	Use alternate interpretation (new)
Ctrl-h	
Ctrl-i	Italics (toggle)
Ctrl-j**	Jump to cell
Ctrl-k**	Insert one braille cell (used when keyboard is in typewriter mode)
Ctrl-l	
Ctrl-m**	Read at cursor (new – speech output)
Ctrl-n**	Find next, Replace next (next in general)
Ctrl-o	Open file
Ctrl-p	Print
Ctrl-q**	Query cursor position (new – speech output)
Ctrl-r	Replace
Ctrl-s	Save file
Ctrl-t**	Display tab ruler (new)
Ctrl-u**	(Underline emphasis (toggle), reserved for future BANA formats)
Ctrl-v	Paste
Ctrl-w**	Toggle between last two kinds of views (new)
Ctrl-x	Cut
Ctrl-y**	Insert last cell again
Ctrl-z	Undo
Ctrl-F3	Previous view (toggle)
Ctrl-F4	Close file
Alt-F4	Close application (e.g., close Braille2000)
Ctrl-spacebar	Insert non-breaking space
Ctrl-hyphen**	Insert breaking hyphen
Shift-Ctrl-hyphen**	Insert vanishing hyphen
Ctrl-tab	Increase style level (e.g., list level 2 from list level 1)
Shift-Ctrl-tab	Decrease style level (g.e., list level 1 from list level 2)
F1	Help for current menu item or dialog box

Nested Lists

In the 2011 Formats specification, nested lists and poetry use a runover position that is two cells to the right of the right-most indent, i.e., all lines of the list (which are distinct paragraphs) must be coordinated for overall correct format, hence you must declare each nested list.

To create a nested list from an ordinary list (and from a single-level list or a pre-2011 nested list), highlight the entire list (just one list at a time), and click Nested List in the Do menu. The list becomes a Block and shows with a light blue background and all runovers are dynamically coordinated. Items in the list can be re-indented as desired (Ctrl-Tab and Ctrl-Shift-Tab are handy during input) and the runovers throughout the list will remain coordinated.

When entering a list, begin entering the first list item and then click the Style button and select N-list and level 1. Press Enter at the end of each list item to extend the list. Use Ctrl-Tab to move the indent to the right; use Ctrl-Shift-Tab to move the indent to the left. (You know it is working when the added material is blue in background color.)

To enter non-list prose, press Enter to start a new paragraph and click the Style button and select the format desired.

Find

The Find dialog box supports “Find”, “Replace”, “Insert Before”, “Insert After”, “Ambig Trans” (Ambiguous Translation).

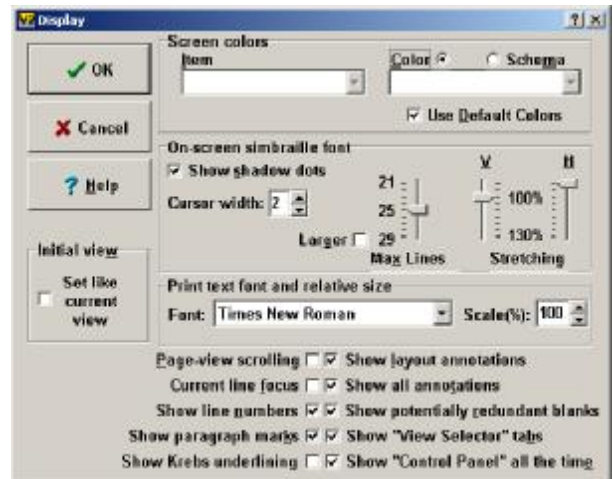
Find	<p>Searches the entire document or a selection for the given phrase (one or more cells or one or more characters or for a pattern match to a regular expression). The search may be modified by “Case insensitive” (for characters) or by “Word” for cells or characters.</p> <p>Top/Cursor/Next highlights one matching phrase starting from the given start point</p> <p>Find All highlights all matching phrases</p>
Replace	<p>Replaces a phrase found by Find or replaces a unit in a selection with the given phrase. If a Find phrase is being replaced that was identified by matching a regular expression, the replacement phrase may use the subexpression notation to form the replacement phrase, else the replacement phrase must be all braille cells or all print characters.</p> <p>Top/Current/Next searches for a phrase matching the Find specification starting from the given start point and then replaces it, or it identifies the next selection unit following the given starting point and replaces it in its entirety.</p> <p>Replace All searches for all matching phrases and replaces each one or it replaces all units in the current selection.</p>
Insert Before	<p>Insert Before operates similarly to Replace, except that the original phrase is retained in full and any new phrase is inserted at the beginning of the original phrase. When the Find phrase is located by matching a regular expression, the new phrase may be composed using sub expression notation.</p>
Insert After	<p>Insert After operates similarly to Replace, except that the original phrase is retained in full and any new phrase is inserted at the end of the original phrase. When the Find phrase is located by matching a regular expression, the new phrase may be composed using sub expression notation.</p>
Ambig Trans	<p>Ambig Trans searches for those braille elements that, although not necessarily wrong, do not back-translate to the prose they represent. This search may or may not be of value. Braille is by nature ambiguous in certain situations. Unlike V1, V2 maintains print text for the entire document while it is open (V1 regenerated print text on the fly on demand as you scrolled through the file), and then when the file is saved as a .B2K type, print that can be regenerated from braille is discarded while the print that cannot be regenerated (i.e., print for those words this search feature will report) is stored in the file. The result is persistence of the original print without enlarging the .B2K file except for the ambiguous elements (which are usually few in number).</p>

Annotation management

Most discretionary document control is handled via “annotations” in V2 very much as was the case in V1. The annotation locations continue to be displayed using small yellow triangles. Various commands (especially “Do” commands) create annotations as needed and as before the annotations usually act and get planted at the position of the cursor and sometimes taking into account the region that is highlighted.

Document control not handled via annotations relates to properties of paragraphs (i.e., “Styles”), and those are handled via the Change Format dialog box and the Style button.

V2 has some new ways you can work with annotations after they have been created, as follows.

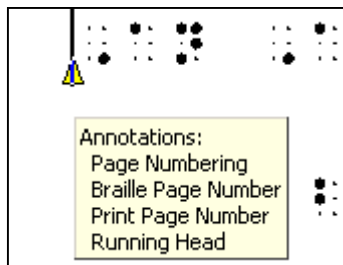


Visibility

The Display dialog box (via Adjust and Display) is identical to the one in V1. There are two options “Show layout annotations” and “Show all annotations”. If both of these are unchecked, the yellow triangles do not show.

Hover

As in V1, if you hover the mouse over a yellow triangle, a message box appears, announcing the kinds of annotations at that spot.



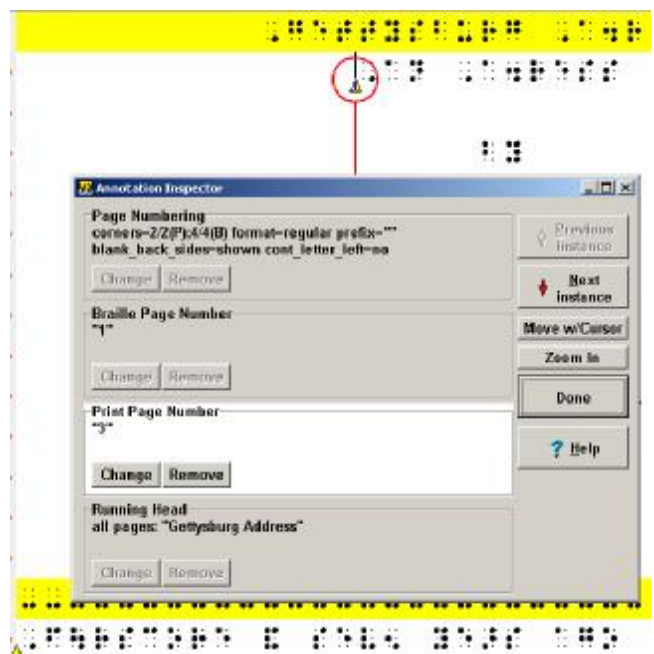
Hover (definition): move the mouse pointer to the spot and then cease all mouse movement for a few seconds (do not click any mouse buttons).

Annotation interrogation and control

With the mouse pointer on a given yellow triangle, click the right-hand mouse button (note that there is no mouse pointer shape change in V2: the arrowhead shape is used for everything). This will display the Annotation Inspector, visually tied to the given yellow triangle.

The dialog box will show one subpanel for each annotation at the given location (in the snapshot, four annotations are revealed).

Click in one of the subpanels to activate that part of the dialog box, and that will provide buttons for “Change” (to edit the properties of the annotation), “Remove” (to eliminate the annotation, “Previous Instance” (to navigate to an earlier annotation of the same type), “Next Instance” (to move similarly to the



next instance), and “Move w/Cursor” (reposition the annotation via arrow keys).

This snapshot shows the above dialog box after clicking in the third subpanel (but not on any button). Now the Change and Remove buttons apply only to the Print Page Number. The Next Instance button will reconnect the Inspector to the next print page number annotation (which happens to lie in this file just below the dialog box at the beginning of the last braille line shown). The Previous Instance button is grayed because there is no previous instance of print page number.

Move w/Cursor

The “Move w/Cursor” button is an on/off toggle, that when “on” activates the arrow keys for dynamically repositioning the selected annotation (Print Page Number in this case).

In this snapshot, Move w/Cursor is “on” and the down-arrow key has been pressed, moving the Print Page Number annotation down to the third braille line. Further motion via arrow keys is possible until Move w/Cursor is turned “off”.

In this case, because the Print Page Number annotation controls the manner of print number expression, the motion caused a page turn indicator line to be generated (and as a result the annotation is on line four because line three is the page turn line).

The three other kinds of annotations are unchanged, still on line two. They could be moved individually if desired.

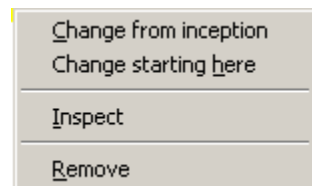
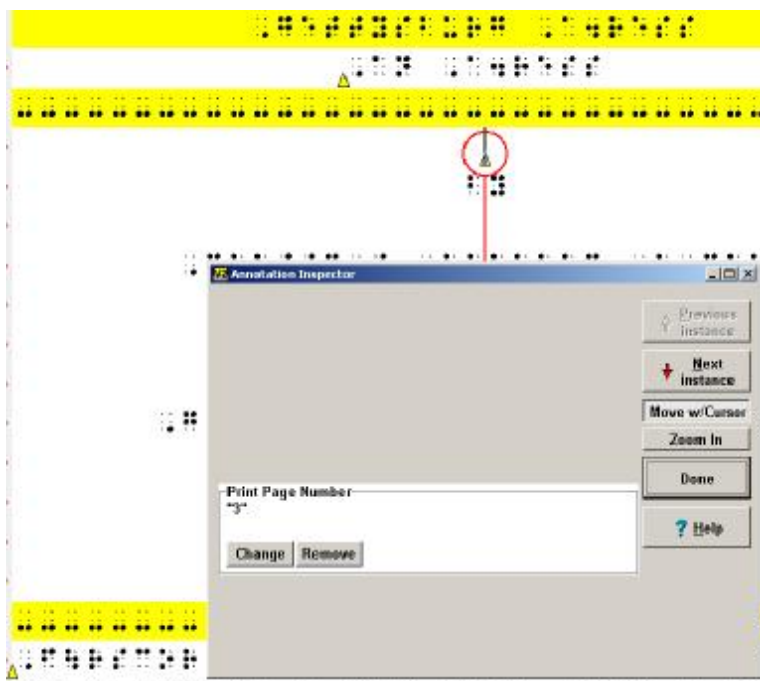
When finished, click the Done button.

If you need to change what the annotation does, you click the Change button. In this case, that brings up the Page Numbering Control, focused on the Print Number only (it is a portion of the Page Numbering dialog box).

You can also adjust page layout annotations without using its yellow triangle, via right-click on the yellow page element that was produced.

In the example shown here, right-click to the page turn indicator line shows the popup menu shown here.

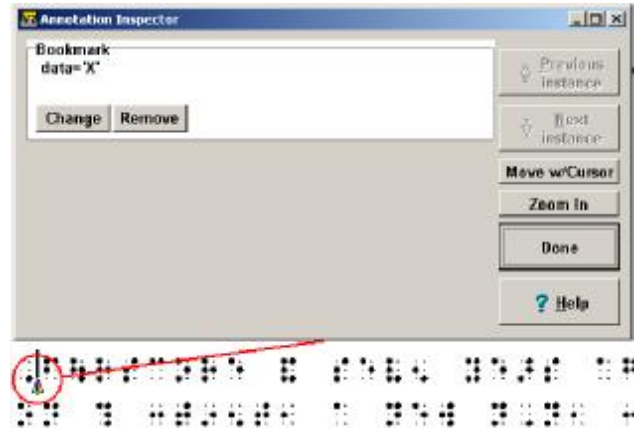
The “Inspect” function displays the Annotation Inspector (you can use this method to inspect an annotation even when yellow triangles do not show on the screen, but it works only for layout annotations, those that generate yellow braille on the page).



“Change from inception” lets you adjust the original settings (for example, correcting all running heads going all the way back to the first page on which they occurred due to the same running head annotation).

“Change starting here will plant an overriding annotation at the current location (for example changing a running head at the current location without altering those that came before).

“Remove” will delete the annotation, possibly affecting many earlier pages. If you want to know where the annotation is that you might remove, use Inspect to see it and then use the Remove button in the Annotation Inspector.



Control panel

The Control Panel is a tool bar that sits to the left of the View Selector (the sidewise-reading tabs that let you select which document to view) which is in turn to the left of the Editing Panel (optional line numbers, paragraph marks, braille or print prose).

The Control Panel can be hidden to reduce the Braille2000 window footprint. (This is done via View and Control Panel.)

New in V2 are the buttons “Tools” “Math” “rtf” “Import” “BkMark” “Insert” “Do” “A” “S” and four small emphasis buttons B I S and U.

The buttons Tools / Math / rtf select what shows in the space of the Control Panel. Because the Math and rtf panel data may be intense, in V2, the width of the Control Panel is adjustable, by dragging the separator line between it and the View Selector right or left.

If you hold down Ctrl as you click the rtf button, the rtf markup panel appears to the right of the Tools panel: you can see them both.

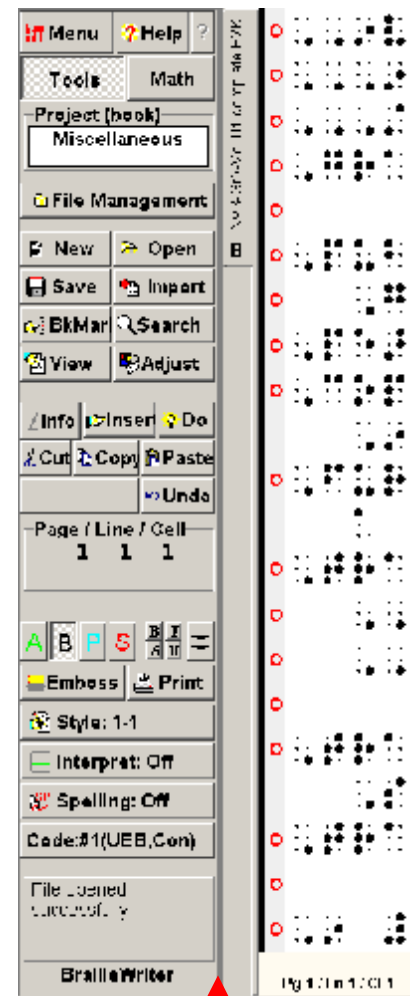
Note that when working with a NIMAS file, the rtf button reads “dtbook” and the panel contains extensive markup data for the NIMAS file.

“Import” replaces both “Insert” and “Append”. The new “Insert” is Insert from the Edit menu.

“Do” functions can now be performed more easily via the Do button (this should simplify how-to tutorials).

“BkMark” does X/Y/Z and general bookmarks.

There are four (smaller) toggle buttons B I S U for bold, italics, script, and underlined.



Adjustable

Boxing Lines

V2 has dynamically-sized boxing lines. Such lines adjust their length automatically to allow for page or line numbers and automatically resize when the document page width is changed.

To insert a boxing line, use the Insert button or use Edit / Insert /Boxing line in the main menu, and then select the type of line you want:



Type	What you get	
Bottom	a line of (1245)'s plus blank line after	
Exterior	a line of (123456)'s	
Thin	a line of (25)'s	
Top	a line of (2356)'s plus blank line before	
Around	top and bottom lines enclosing what was highlighted	
	0 dots (remove)	boxing lines removed
	1 dots (thin)	(25)'s before/after
	2 dots (regular)	(2356) before, (1245) after
	3 dots (exterior)	(123456)'s before/after
	4 dots (exterior/thin)	(123456)(25) before (25)(123456) after
	5 dots (exterior/regular)	(123456)(2356) before (1245)(123456) after

Note: not all patterns have a role in Formats

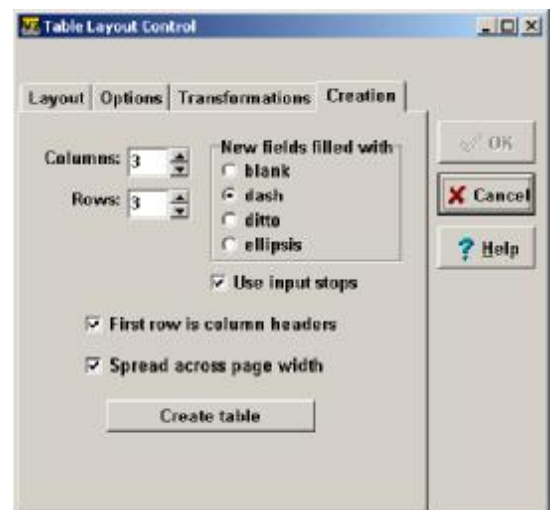
Tables

To insert a new table, click Insert and select Table. You get the dialog box to the right. Set the number of columns and rows and click "Create table".

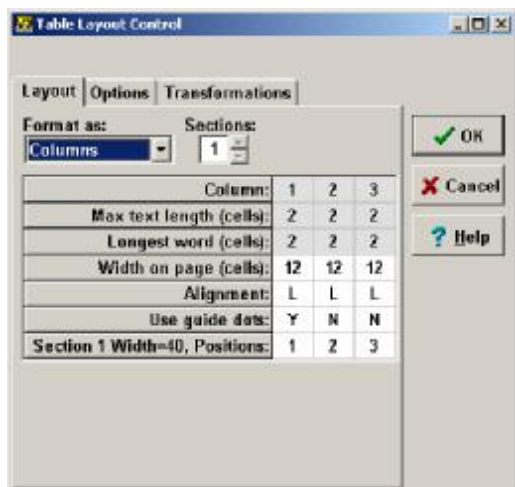
The number of columns and rows is based on the print copy and the top row (column headers counts as a row). If the first row is column headers, put a checkmark in "First row is column headers" and you will get a separator line (in columnar format), otherwise there will be no separator line.

A new table of the given dimensions will be created with each field set as per the "New fields filled with" setting.

If you are *not* creating a table that will remain empty, you probably want the "Use input stops" option that will

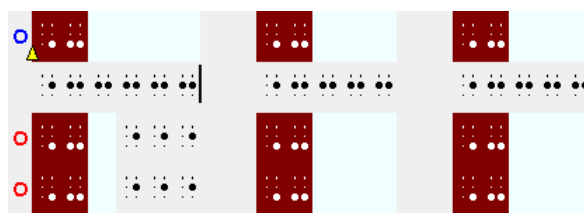


create an input stop at each field (use the Tab key to visit each input stop, one at a time, and enter data).



After you click “Create table” you will see the Layout panel as shown at the left. For small simple tables, just click OK.

An empty table of the dimensions you gave will be inserted into the file. The rust-colored areas (nine in this example) are the input stops, one in each empty field of the table. The tab key selects and highlights one input stop at a time, moving through all of them in a circular fashion. For data entry, you press tab, an input stop is highlighted, and you braille the value of that field. Then press tab to go to the next field until the table is all filled in.



You can also fill in a table in Print View, just the same way.

The background color of a table is light blue, because it is a “block” with special behavior. Other block elements in V2 are nested lists and drawing spaces.

Note that the filler (dash/ditto/ellipsis) is code dependent.

Editing Via Keyboard Entry

In the ASCII view and the Braille view, keyboard input is naturally six-key. You can change this to ASCII-braille typing via Adjust and Keyboard. In the Print view and the Source view, keyboard input is always by typing. You can enter a single cell or trigger a Speedbraille™ key by entering Ctrl-k followed by a single six-key cell or by a Speedbraille trigger (as in normal six-key mode).

Pressing the Enter key when the cursor is in the middle of an existing paragraph introduces a paragraph break at that location, splitting the paragraph into two adjacent paragraphs. Both paragraphs have the same indent and runover settings as before. If you split a paragraph in the ASCII view or the Braille view, the braille cells split at the cursor and the fragments are each back-translated into print according to the language setting. If you split a paragraph in the Print view or the Source view, the print letters split at the cursor and the fragments are each translated into braille according to the language setting. If you split a word or phrase that has print emphasis, both fragments have the same emphasis and the resulting translated braille fragments will each manifest this emphasis. Splitting a word in the ASCII view or the Braille view does not replicate composition signs and if the word split was italicized, the first fragment would continue to be italicized (it would keep the italics indicator) while the second fragment would not. Thus there are subtle differences with the behavior of the Enter key between viewing modes.

If keyboard entry is performed while a single phrase is highlighted, the highlighted material is first deleted and then the keyboard cell or letter is inserted in its place. If a multi-highlight is showing (i.e., a multi-part selection), keyboard input merely generates beep sounds and nothing is either deleted nor entered. You may delete all elements of a multi-part selection using the Delete key and after that is done keyboard input will be inserted at the location of the cursor.

Page Control (braille page break and more)

You will find Page Control in the Do menu. One function of Page Control is to manage Page Break. The choices “To next front page” and “To next back page” are available only for interpoint files (use Adjust and Document to set the embossing mode).

New in V2, you can break to the next page and suppress the running head (think of the first text page in a volume). The attributes currently grayed out will be working in a later release.



If you click the “Insert blank page tab” you will see the second panel, shown to the left.

This function inserts a given number of blank pages (in an interpoint file, a number of blank sheets, blank on both front and back). You can specify a number from 1 to 9 and you can cause certain page layout elements to be suppressed on the blank pages. To insert one blank page (one blank front/back pair in interpoint), you can click the new “Insert” button and select “Blank Page”.



Drawing Spaces

The drawing space feature is pretty much like that feature of V1, however some aspects of it are being simplified (the feature in V1 would sometimes get confused... it was too complicated). To insert a single whole-page drawing space, click the “Insert” button and select “Blank Drawing Page”.

Source Files (RTF files)

You open an RTF file via the “Source file” option (from the Open button). You then select the braille code you want to use and Continue to an Open-type dialog box. The current release supports two kinds of source files: rtf and xml (NIMAS). Source files by nature have an internal “markup” mechanism of some kind. Borrowing from the formalism of xml, the kind of markup is identified by a “Doctype” (document type) identifier that announces the semantics of the markup mechanism. The Doctype identifier is what appears on the button at the top of the control panel, i.e., “rtf” or “dtbook”.

Source file type	Filename suffix	Doctype name
Rich Text file	.rtf	rtf
NIMAS file	.xml	dtbook

Other source file types will be supported in the future.

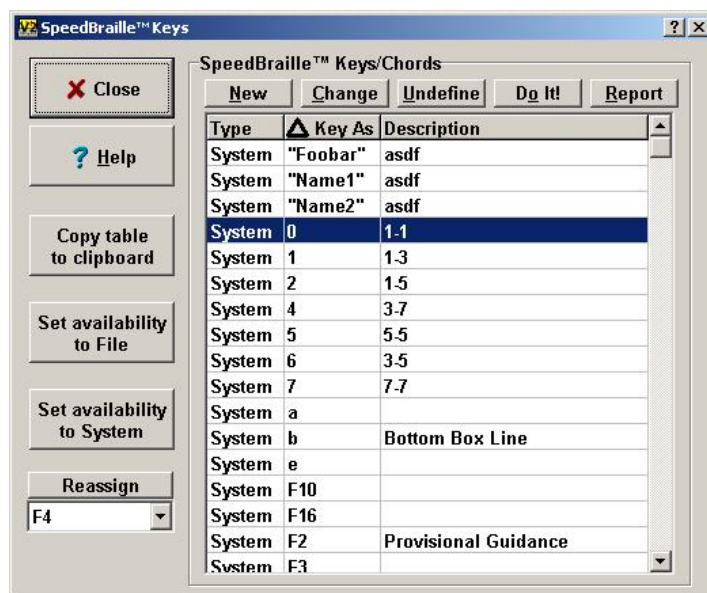
After you choose which RTF file to process, Version 2 creates a new braille document, sets the language attribute, opens the file, translates it, and shows it on the screen. The filename suffix will be .B2K, the new type of standard output file created by Version 2.

When source files are opened, there is a document-type-specific source “panel” that is available on the left. Version 2 has several panels (toolbars) that share the same space on the left. “Tools” is the familiar assembly of handy buttons. “Math” is part of the new (not-yet-available) “math tools optional feature” (the Math button is grayed out for the moment). When you open a source document with a Doctype (document type formal specification), you get a markup panel named for the document’s Doctype, namely “rtf” for RTF files; and “dtbook” for NIMAS files.

Rich Text data can also pass into Braille2000 from the Windows Clipboard.

Speedbraille™ Keys

V2 offers Speedbraille™ Keys but their nature is somewhat different. A major change in the internal mechanism was mandated by security changes and bugs in Windows 7 & 8 & 10. In V1, Speedbraille keys (a kind of “macro”) were done by recording and reusing keyboard data, literally as if a robot with lightning input speed retyped the definition whenever you triggered a Speedbraille function. It was because of this basic nature that mouse effects did not “record” (you had to do menu functions via the keyboard). That has all changed.



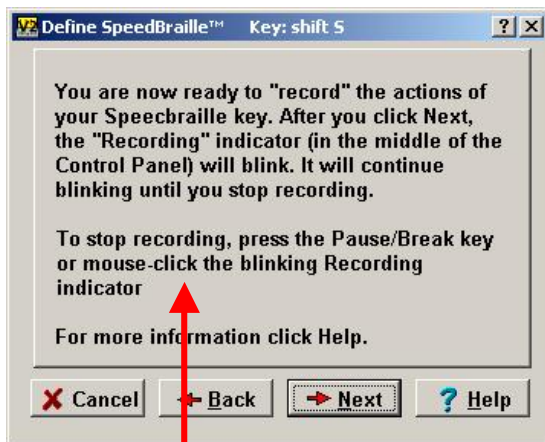
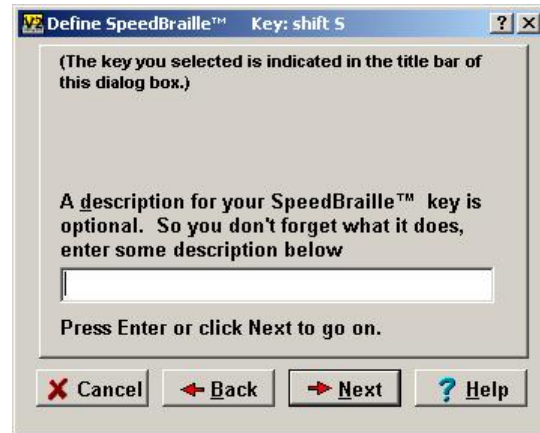
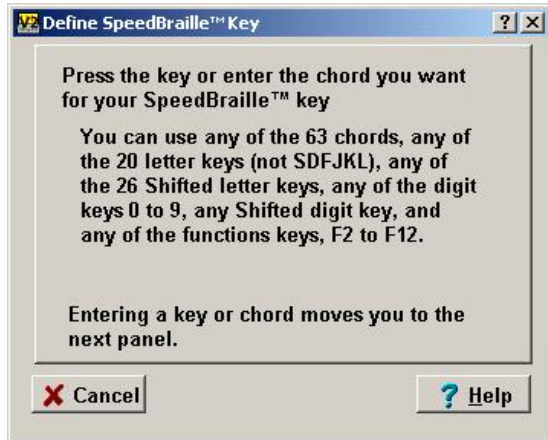
V2 does not record keyboard data as such. Instead, V2 generates internal transactions and records those. Each cell of six-key input is a transaction, as is each use of a menu function or dialog box. Because the system is generating and storing transactions, the V2 Speedbraille mechanism can “record” mouse-generated actions (such as menu use and dialog box setup). This gives a more natural behavior. But... the kinds of information recorded in the definition of a V2 Speedbraille key are totally different, at least for now, there is no cross-compatibility: no loading of your old V1 Speedbraille definitions (sorry!). You will need to re-record them to make them work in V2.

Also, there are a few menus and dialog box effects that currently do not generate transactions and for them there is, for the moment, nothing to record. If you hit one of those situations (unable to “record”, please report it). The goal being to be able to record any set of actions and package them into a Speedbraille key definition.

The Speedbraille procedure looks essentially the same. **Click View and select Speedbraille Keys to get the Speedbraille Keys listing** (above):

Then click New to begin defining an new key or to change an old Key.

The sequence of panels is the same:



While “recording” you can now do things with either the mouse or the keyboard.

Note that you can stop recording via the mouse by clicking the "Recording" indicator (it V2 that is in the upper-left corner of the screen).

If you look at a “report” you will see cryptic transaction statements rather than cryptic keyboard data. V2 also as a “scripting” mechanism that (in the future) will let you combine transactions (the same kinds as which make Speedbraille macros) into stored routines (a better place for seldom-used but important macro behaviors). There is also an intimate relationship between V2 Speedbraille macros and V2 user-defined paragraph styles.

New buttons (see the Speedbraille Keys dialog box above)

Copy Table to Clipboard Use this to put the Speedbraille listing on the clipboard, to paste into WORD or Wordpad to print out for reference.

Set Availability to File Click one or more entries (to highlight item) and use this button to change the storage class of all items at once.

Set Availability to System Click one or more entries (to highlight item) and use this button to change the storage class of all items at once. To import V2 Speedbraille definitions from another computer, Save a .B2K file at the originating computer, then open that file at the target computer, view the Speedbraille Keys listing, select the File type entries you want to grab, and use this button to change their storage location to "System".

Reassign Use this to change the trigger key for one selected item. Click the down-arrow to view a listing of the possible trigger keys. In that list, chords are listed by dot numbers in parenthesis (a "chord" is six-key input in which dot keys and the spacebar are pressed simultaneously).

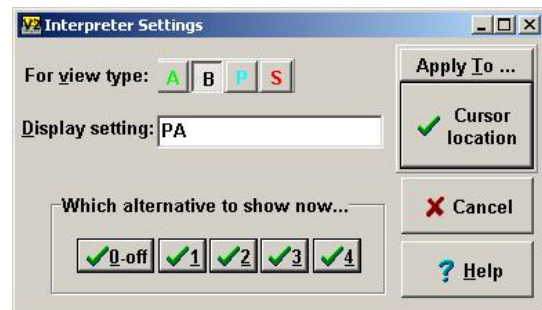
It is not uncommon to have Speedbraille macros to effect paragraph styles. Such macros behave exactly like custom paragraph styles (defined via the Paragraph Style dialog box), and in V2 they are in fact the same thing... you will see custom paragraph styles in the Speedbraille list.

If you select such an entry (a macro that only effects a paragraph style), then in the list of trigger keys, a final entry, "StyleName" will show. This is not actually a trigger key but is the way to access the stylename property.. Select "StyleName" and then a name box will show up. Enter a new name to have this function show (in the Style selector popup) as a custom style. You can use this function to convert between a named style and a key-triggered Speedbraille macro that does that style.

Interpreter line (green line)

V2 has four ways of looking at prose: Ascii, Braille, Print, and Source. Print and Source are the same except for constraints on the vertical dimension (print view maintains the vertical spacing of the braille).

Correspondingly, the green line has four display options, A, B, P, and S (in the green line P and S are identical). By right-click to the Interpret button, you get the dialog box shown here.



Into "Display setting" you can put one or more display type letters (ABPS) in the following ways:

- A single letter yields a single-line green line.
- Two or more letters written together (such as BA) produces a green line with as many lines of prose as there are letters, one line of each kind, e.g., Braille followed by Ascii.
- Multiple such expressions may be given separated by commas, and each element of such a setting represents one of up to four alternative behaviors. The Ctrl-g key and mouse clicks to the Interpret button will cycle though the alternative formats. For example, BA,PA has two alternates each with two lines: Braille and Ascii, or Print and Ascii.
- There are separate settings for each of the file view types, i.e., there are separate green line behaviors for when the screen shows Braille and when it shows Print. For example, you might like the setting P for a Braille View (main screen braille, green line print) at the same time as a setting of B for a Print View (main screen print, green line braille). There is a separate setting for each of the main screen views (Ascii, Braille, Print, Source).
- All of the above settings are stored per language, i.e., there can be different green line behaviors for English versus Nemeth.

The five checkmark buttons in the dialog box are for the immediate selection of “Off” or one of the four alternative modes. Clicking the Interpret button cycles through the modes (if the setting defines multiple modes), Ctrl-g cycles through the modes, but via the buttons 1, 2, 3, 4 you can directly switch to a particular mode. You can make a Speedbraille key to select a particular mode.

Below is a green line with the setting PA. The setting AP would put Ascii on top.

The image shows a Braille editor interface. On the left, a line of Braille is shown with a green highlight. Below the Braille, the text "Liberty, and dedicated to the" is displayed in a serif font, also highlighted in green. Below that, the text ", LIB]TY1 & D\$ICAT\$ 6!" is displayed in a monospace font, also highlighted in green. To the right, a dialog box titled "Change emphasis" is open. It has an "Apply to..." section with "Current paragraph" selected. The "Number of operations:" is set to 1. Under "Operation 1", "everything" is selected in the "is" box, and "italics" is selected in the "becomes" box. The "is" box also contains "plain" and "bold". The dialog has "Cancel" and "Help" buttons.

Emphasis Control

By default, source file emphasis (italics, bold, not yet underlining) is retained during import. There is a new way to adjust emphasis after import: click Adjust and select Emphasis.

The box to the left of “becomes” selects the existing emphasis setting. The box on the right selects what you want the emphasis to become. The box on the right can select one of plain, italics, bold. The box on the left lists all the kinds of emphasis that are in the selected application scope, which can be "paragraph", "highlighted text", "rest of file" or "all of file". The scope can be changed via the "Apply to..." button.

Emphasis changes during the importation of RTF files can be established using percent directives, described below.

More About RTF Files

You open an RTF file via the “Source file” option (from the Open button). You then select the braille code you want to use and Continue to a Open-type dialog box. (The current release supports two kinds of source files: rtf and xml (NIMAS)).

After you select the RTF file you want to process, Version 2 creates a new braille document, sets the language attribute, opens the file, translates it, and inserts the braille into the new file. The filename suffix will be .B2K, the new type of output file created by V2.

You can also Copy RTF data to the clipboard and Paste it into an existing file. In Version 2 you do not need to use “special paste” (it can usually figure out whether the clipboard contains braille or print, and it will translate print to braille; if the clipboard is such that it is unclear whether the material is braille (which comes in as-is) or is print (which is to be translated), Version2 will ask you.

In the markup panel (i.e., dtbook or rtf) each line is a “node” in the style hierarchy. Each line is linked to the relevant location in the braille text. “p” denotes “paragraph”. For example this document begins with two paragraphs in Normal style followed by one paragraph in heading_1 style. By clicking on a node name, the cursor will jump to the corresponding location in the braille panel (which is called the Editing panel) *depending on the current “cursor transfer” setting, described below.*

Using the Markup Panel for Summarization, Navigation, and Control

Summarization

Via the Options button and the Reveal selection, you can add to the nodes in the Rtf panel a variety of useful information, such as the print text or the print page number (even both at the same time if you wish). Remember that RTF files do not of themselves support print numbers (you would need to augment such a file with %pn= directives and perhaps take advantage of page breaks).

You can change the width of the toolbar by dragging its right-hand border left or right. If you use Reveal to show auxiliary information, you may need more room. A new wide-screen monitor or notebook helps give extra width.

Navigation

When you click on a node in the Rtf panel, the cursor jumps to the corresponding spot in the Editing panel. If you hold down the Ctrl key as you click, the corresponding segment of the document becomes highlighted.



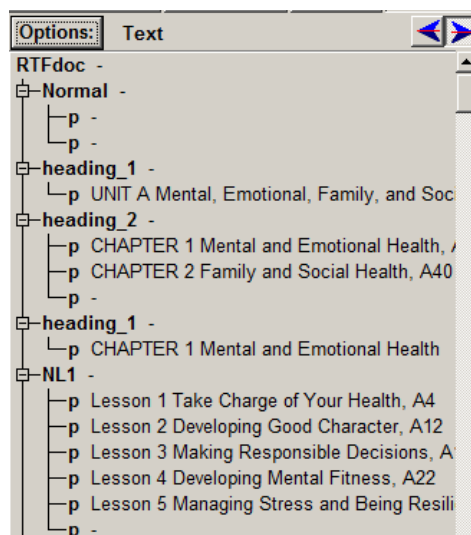
You can also do the reverse: click in the Editing panel and have the Rtf panel indicate (via bold linking line) the corresponding node in the Rtf hierarchy. These behaviors are controlled by the “cursor transfer” buttons in the upper right corner of the Rtf panel. There is one button for right-to-left transfer (Editing panel position to Rtf position) and another for left-to-right transfer. The buttons toggle.

Control

You can use Ctrl and clicking in the Rtf panel to select multiple nodes (representing multiple regions of the file, as you click to select in the Rtf panel, corresponding text areas become highlighted in the Editing panel). You can also select nodes based on node name and/or node depth.

Having established a multi-selection, you can do things to the selected text, such as change style, set guide word text, format contents entries, etc. This notion of “select and act”, using multi-selection capability, is unique (in braille processing) to Braille2000.

You can form a multi-selection in the Editing panel using the mouse to highlight text, while holding down the Ctrl key.



Import control via Translation Typing Directives

“Translation Typing Directives” are transcriber-added cues for the Braille2000 import mechanism for RTF files. They act as a sort of additional markup. Each directive begins with the percent sign, and thus they are also called “percent directives”. Their use is optional. Note that print text coming from the Windows clipboard is also RTF and thus these directives can also be applied via the clipboard.

There is only a vague relationship between the page layout of a typical print document and the standard layout of braille pages. In most cases, at some point human judgment is needed to establish the desired braille page arrangement. Such layout changes can be done (a) to the braille after importation into Braille2000, or (b) to the print file, in ways (unique to Braille2000) that will import with good braille page layouts.

There are five text to braille layout relationships which can be used (besides also using percent directives). **These relationships guide the braille format when doing implicit format imports.**

1. Centering versus left-justified

Text which is centered generates centered braille. Text which is Left-justified generates normal (left-justified, non-centered) braille. Other text alignments, such as right-justified should not be used.

2. Italics versus bold versus normal font

Emphasis in print is transferred to the imported braille.

3. Indent and Paragraph (left) Margin

The amount of paragraph indent and “runover” (in text, one uses the term left margin, but it is the same concept as runover) determines the indent and runover of the resulting braille. The text indent and left-margin settings should be set in jumps of quarter inches, with each eighth-inch corresponding to one braille cell position. The table below shows the indent and left-margin settings for popular braille paragraph forms. Note that the left-margin measurement is taken from the left side of the printable area on a page and thus does not include the page margins.

Braille form	Text indent	Text margin
1-1	0”	0”
1-3	0”	0.25”
1-5	0”	0.5”
3-1	0.25”	0”
3-3	0.25”	0.25”
3-5	0.25”	0.5”
5-3	0.5”	0.25”
5-7	0.5”	0.75”

4. Tabs

Tab stops work much the same way, with each 1/8” of text space mapping to one cell position.

5. Choice of font

You may use any font you like, with the following two provisions:

- a) Elements of text that should be set in Computer Braille Code can be set in Courier font;
- b) Elements of text that are literal ASCII-braille symbols can be set in a simbraille font, such as “Braille from BRL2000.

All other fonts are translated to the target language, set in advance of translation via the Braille2000 "Language" button. See also the %cbc= and %brl= directives.

You can do an entire file using implicit formatting (without percent directives affecting indent and runover), as described above. As an alternative, Translation Typing offers several directives, listed below. A directive is placed alone on a line or at the beginning of a paragraph (it is followed by one space if not on a line by itself). It specifies an overriding braille format for the immediately following text and all subsequent text that has the same text layout (same print-implied indent and left-margin, and same style (centered versus left-justified)). When the text layout changes, implicit formatting resumes, unless (a) you use another format directive, or (b) you have used %explicit to switch to “everything-is-explicit” mode. In “explicit” mode, braille layout is determined *exclusively* by directives.

When explicit control is used throughout the document, text paragraphs might as well all be blocked at the left margin (0” indent and left-margin).

There are thus three ways to prepare Translation Typing text:

1. Implicit Layout (the default behavior)
by carefully laying out the text, making sure the indent and left-margin are right for braille,
2. Hybrid Layout (using directives as needed)
by laying out the text in a good-looking way, and using a format directive whenever additional control is needed, and
3. Explicit Layout (put %Explicit at the top of the document)
by leaving prose layout as-is and indicating all braille layouts via directives.

In the hybrid approach, a directive that affects indent or runover applies to the following print paragraph and all subsequent print paragraphs that have the same *print* layout. When the print paragraph layout changes, the effect of the directive terminates (but if using %explicit, the effect endures until the next layout directive that changes indent and/or runover).

Layout Options (new)

Braille2000 V2 does not use “Rich Text Options” or “Print-text Options” dialog boxes. Instead, those controls are specified using percent directives, as follows: (* denotes new capabilities)

Main Document Style

Blocked	%explicit %1-1 %tb
Indented	%explicit %3-1
Translation Typing	(assumed)

Options

Generate Running Head	%runhead (placed immediately before one paragraph of running head text)
Cancel Running Head*	%runhead=off

Generate Print Numbers	%pn=off (to suppress print number synthesis)
Next print page at page break	%page=npp
Preserve hard page breaks	%page=bpb
Ignore hard page breaks*	%page=ignore
Courier font gives CBC	%cbc=courier
Braille font is literal braille*	%brl=braille
Preserve blank lines	%pbl or %pbl=off
No paragraphing blank lines	%tb or %tb=off

Each of these directives is defined below. With the new approach, these settings can now change (if desired) within the RTF file (in V1 such settings always applied to an entire RTF file). You *can* paste with preset directives: see also “Special Paste” below.

Directives you can add to the RTF file (“percent directives”)

There are a few aspects of page layout and translation that are unique to braille. These are indicated by special directives which can appear inline (in the text) or as a separate paragraph. A special directive is a single word that begins with the % sign. When written inline (with the text) it is typed as a word, set off (as needed) from any adjacent text by blanks. The following special directives are defined. Italics denotes a variable (fill-in) part of the directive.

Directive	Use
%explicit	To ignore paragraph shape and rely only on directives for braille format control
%implicit	To honor paragraph shape (indent and margins) and to format the corresponding braille paragraph according to the scale of 1/8” per cell (you may use directives below as required but some effects wear off)
%block	To set the following paragraphs as 1-1 braille paragraphs
%blockv	To set the following paragraphs as block paragraphs of level v
%box= <i>letters</i>	To insert a boxing line. Letters are TMB (top/middle/bottom) optionally with E (exterior), such as %box=T or %box=ET (note ET and TE act the same)
brl= <i>keyword</i>	To take print as literal ASCII-braille when <i>keyword</i> is part of the fontname (default setting is brl=Braille)
cbc= <i>keyword</i>	To translate print into computer braille code when <i>keyword</i> is part of the fontname (default setting is cbc=Courier)
%center	To cause the single following paragraph to become centered braille
%centered	To cause all following paragraphs to become centered braille
%emp= <i>b</i>	To map print emphasis (<i>p</i>) to braille emphasis (<i>b</i>) <i>p</i> and <i>b</i> stand for one or more of the letters ibu (italics bold underlined) used singly or in combination without spaces. The letter p (plain) may be used alone for braille emphasis. When fewer than three letters are used, the emphasis not mentioned is implied: for example i used alone implies also ib iu ibu (i.e., all forms of italics) whereas iu implies also iub (all forms if underlined italics). In some cases, you may need to use multiple %em directives to get the mapping you want, and then the order of such directives may be important: for example %emu=i %embu=p together in this order would cause all print underlining except bold underlining to be braille italics.

	Unless otherwise mapped, other patterns of emphasis are the same in braille, for example, in this case non-underlined bold would be bold, etc.
%gv	To change the language (braille grade) to number v, where v is 0: literal ASCII-braille (source text is not translated at all) 1: uncontracted form of the current language code (if such is possible) 2: contracted form of the current language code (if such is possible) Note: for a document in a foreign language (always uncontracted), 1 and 2 do nothing
%head	To cause the single following paragraph to be centered with a blank line before and after
%indent	To set the following paragraphs as 3-1 braille paragraphs
%indentv	To set the following paragraphs as indented paragraphs of level v
%L	To set the language back to its previous value (if any)
%left	To cause all following paragraphs to be regular (uncentered)
%list	To set the following paragraphs as 1-3 braille paragraphs
%listv	To set the following paragraphs as list items of level v, i.e., %list1 for the first level, %list2 for the second level. This forms a multi-level list (as per 2011 formats) until you use a non-list directive (such as %indent).
%Lkeywords	To set the language via keywords The same behavior as two items down, but without defining any language number.
%Lv	To set the language to language number v Note: the given number must have been defined previously (see below), or else you will be prompted to select the definition during translation. v must be a positive number (%L0 and %L (no keywords) are equivalent and will reinstate the previous language).
%Lv=keywords	To set the language (for automated translation) Keywords separated by commas and without any spaces are used to set the language for translation and define that choice as language number v so that this same language can be selected throughout the remainder of the document by number only.
%note	To set the following paragraphs as 7-5 braille paragraphs
%off	To turn off all directive processing in the remainder of the file This directive should be placed on a line by itself. With this setting, % is translated as an ordinary character. And of course there is no way to reestablish directive processing (because all subsequent directives are treated as merely plain text). Don't use this just to get % as prose (change a leading % to %% to have it taken as itself).
%page	To generate a braille page break
%page=npp	To do "next print page" at each hard page break in print
%page=bpb	To do "braille page break" at each hard page break in print
%page=ignore	To do nothing at each hard page break in print
%pbl	To start preserving blank lines in print (as blank lines in braille)
%pbl=off	To stop preserving blank lines in print (the default)
%pn=number	To give the print page number The number may have a prefix such as Ref1 or Intro1. If the prefix looks like a range such as 3-1 (chapter 3 page 1), add :1 and enter 3-1:1. If given as a range of page numbers such as 2-3, it is taken as the notation for included empty pages. If given with an ending :2, it is taken as spanned pages

	transcribed as a whole, as in 2-3:2. If you must give a braille continuation letter in print, add it with a colon, as in %pn=43:b (the print number would be b43), otherwise b would be taken as a prefix.
%pn=off	To cancel the generation of print page numbers
%reqsp	To check for a blank line in braille at this point, generating one if necessary
%runhead	To cause the single following paragraph to be taken as the running head
%runhead=off	To cancel the use of a running head
%space	To generate a blank line in braille
%spacev	To generate v number of blank lines in braille (v=1,2,...9)
%subhead	To cause the single following paragraph to be a 5-5 paragraph with preceding blank line
%tb	To start "tight blocked paragraphs" (without the traditional blank line between blocked paragraphs)
%tb=off	To stop "tight blocking" (i.e., use traditional format, the default)
%tnl	To generate a left-of-text (beginning) TN symbol
%tnote	To set the following paragraphs as 7-5 braille paragraphs with leading and trailing blank lines.
%tnr	To generate a right-of-text (ending) TN symbol
%toc	To set the following paragraphs as contents entries
%tocv	To set the following paragraphs as contents entries of level v (v=0,1,2,3, etc.)
%toc=off	To end special handling for contents entries
%x-y	To set the following paragraphs as x-y braille paragraphs (x, y are numbers for indent and runover, respectively)
%%text	To use the % character as plain text, use %% to represent %

Importing the RTF file

In Braille2000 V2, you can use Open / Source File to import an entire RTF document and create a braille version of it. You can still use Insert or Append to import by adding the translated text to an existing braille file. You do not have to create a new empty braille file first: you can simply Open the RTF file as a Source File.

You can also paste text from the clipboard by clicking the Paste button. Via right-click to the Paste button you can import RTF material using customized options (see Special Paste, below).

Using the %L directive

The %L directive is needed only in RTF material that is composed of multiple languages, such as a mix of English and French, or where portions are contracted and others are uncontracted (for switching between contracted and uncontracted translation, the %g directive may be used). The directive may be used in defining "special paste" to set the code. Note that "L" stands for "language" although in Version 2 the term presently used is "code".

The %L directive (written here with capital L for clarity, but like other directives it is not case sensitive, i.e., you may use %l just as well) can be used in two main ways:

1. To declare the language for major sections of the document, as in
 - %Lenglish,contracted
 - block of text in English
 - %Lspanish

block of text in Spanish
%Lenglish,contracted
continuing in English

2. To declare a phrase within other prose that should be translated in a foreign language (note that in many contexts isolated foreign words are transcribed with dot-4 notation (i.e., as English) and do not use the foreign language codes), consider
Then he said, %Lfrench Sacré Bleu %L and hung up.
In the above line, the %L by itself returns to the prior language (presumably English).

You identify your choice of language by writing %Lkeyword,keyword,keyword... using as many keywords as needed to make your choice clear. No spaces are permitted in the entire %L directive. If you write %Lenglish, you may sometimes get unanticipated results depending on context because Braille2000 has several languages whose name contains the keyword English. %Lenglish,iceb,contracted or %Lenglish,iceb,uncontracted are better.

There are some built-in biases for selecting the code to use: (a) if both contracted and uncontracted options are available (and you don't indicate which in your %L directive), the contracted code will be used. If your document uses UEB and you use %L to switch to a foreign language and you do not indicate which code group to use, the ICEB group will be used (the same group as for the previously selected code). Similarly, if your document uses EBAE and you use %L to switch to a foreign language with specifying a group, the BANA group will be used.

As Braille2000 operates, you may see code keyword abbreviations in the Code button, such as "con" for "contracted". Those abbreviations may also be used in the %L directive. You will also see that the codes used in an existing file are numbered (#1, #2, etc.). When importing more RTF material into that file, you may use %L#n to re-select the n-th code already in use in the target file.

Using the Code dialog box (right-click the Code button to see it), you can name your own codes (so-called "Quick codes") as synonyms for a system code. For example, if you define "EBAE" as your own code name, you can use %L"EBAE" to select it from within the RTF document (assuming that you have already defined that name). Quick code names (your assigned names) are always written within double-quotes, as shown.

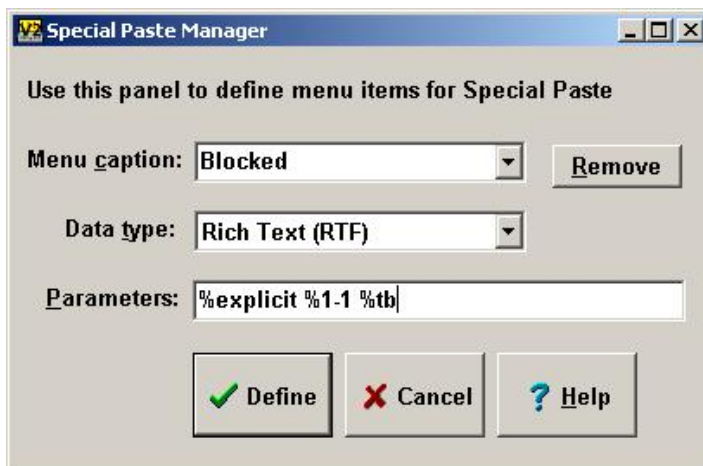
When a document contains a great quantity of language changes, you may define a language number to minimize the directive. Language numbers run from 1 up (and need not be used in sequence, i.e., you can use 23 without having used 1 through 22). A language number must first be defined via the directive %Lnumber=keywords, as in %L3=English,Uncontracted. This directive sets the designated language and defines it as language 3 which may then be used simply as %L3. Once the definition is made, you can use %L3 wherever you would use the longer directive, for example in %L3 a phrase that is to be uncontracted %L and then normal material.

If you use %Lnumber without ever defining what that language number is to be, you will be prompted to select a language to define that language number, while the translation is in progress. Once defined, %Lnumber (for the same number) will indicate the given language throughout the document, during translation. Once translated, language numbers are discarded from the braille file, but language transitions, affecting back translation and spelling checking, endure, and can be spotted as "Lang" annotations.

When preprocessing a print document with comprehensive tools, such as MS Word, you may be able to construct time-saving macros to insert %L directives in an automated or semi-automated manner.

Special Paste

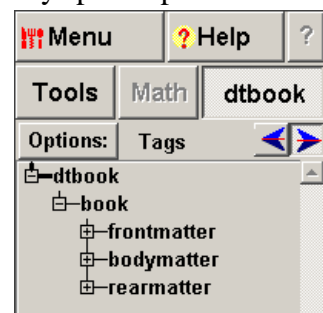
If you right-click the Paste button, you get the “special paste” menu of choices. If you select “Define RTF Paste” you will get the dialog box shown here by which you can name additional special paste menu choices and give an initial line of percent directives to pre-condition the behavior of the import process. For example, you could create the choice “Blocked” (as shown to the right) that makes simple blocked paragraphs suitable to prepare the label for the spine of a bound volume. The codes %explicit %1-1 %tb are given, that (in the absence of contrary codes in clipboard data) would form simple 1-1 paragraphs without separating blank lines.



You could also have special paste choices preset for foreign language material by giving the appropriate %L directive. If you give multiple directives, leave one space between them (just as you would do if you put the directives into the RTF file itself). The parameters are handled as if they constituted an implied first line of the RTF file. You may define as many special paste behaviors as you wish.

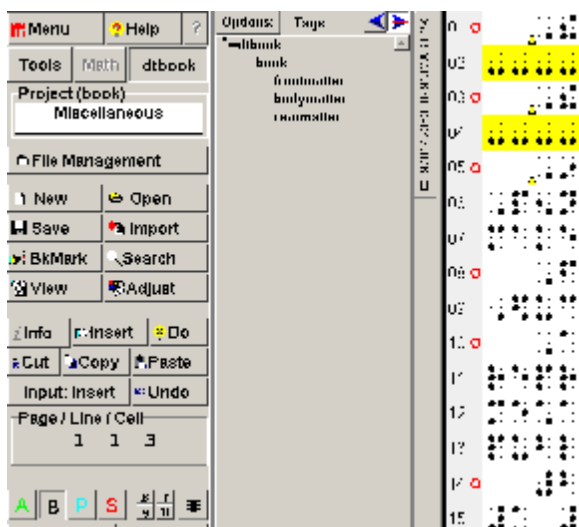
NIMAS Files

The dtbook button shows the NIMAS markup panel. (dtbook is the DOCTYPE for all NIMAS files). Had you opened an RTF file, the button would read "rtf" and would show the "rtf" markup panel. The Tools button shows the "Tools" panel, the conventional Braille2000 set of control buttons. You can adjust the relative widths of the panels by moving the gray divider



bar left or right with the mouse (see the small (red) upward-pointing arrow above).

If you click "dtbook" (or "rtf") while holding down the Ctrl key, you get a two-panel display as shown at the left. The "Tools" panel is on the left and the "dtbook" panel is to its right. Of course when you have two panels on the screen, the editing area is reduced in width.





There are four ways to view the NIMAS document, using the four buttons shown at the bottom of the snapshot at the right: A (ASCII-

braille); B (braille); P (print); and S (source). ASCII, braille, and print always show braille page geometry. Print view items that would distort the braille page geometry are shown as simbraille cells, for things like tall fractions or tall math expressions, and source images. The source view shows document images (if they are available via link) and all math expressions, but because of size differences, the source view may not show all lines on a braille page (you can use vertical scroll to see what does not show initially).

Via Ctrl-w you can toggle between the last two view settings. If you only need the "Tools" panel to change the View, then Ctrl-w can do this even when the Tools panel is not showing.

Navigating using NIMAS tags


The dtbook panel first shows like this (see right). Most of the panel is used to show the tags hierarchy, initially just "dtbook" and "book". Click on  to reveal item detail. Click on  to hide detail. When you click on a tag name (not on the + or -), the editing panel will scroll to the corresponding location in the text, depending on the cursor-transfer buttons.

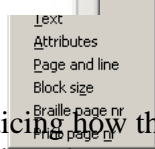
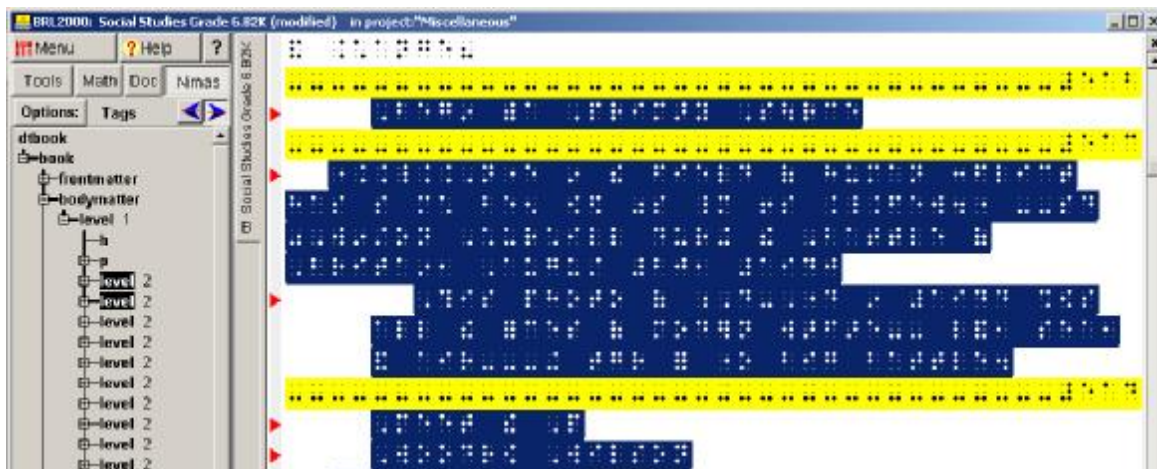


The cursor-transfer buttons show either "up" (inactive) or "down" (active). A click changes the buttons state.

Clicking in the dtbook panel sets the focus node (a bold line traces the path from the focus node to dtbook). When the right cursor-transfer button is down (active): the click will also scroll the editing panel to the corresponding place. Clicking in the Editing panel moves the cursor. When the left button is down (active), the click will also scroll the dtbook panel and open nodes as necessary to set the focus on the corresponding node. You can set either or both cursor-transfer buttons to the up (inactive) position when you want the Editing panel cursor and the dtbook panel focus to operate independently, for example, so that the dtbook panel does not scroll while you are doing editing via the Editing panel.

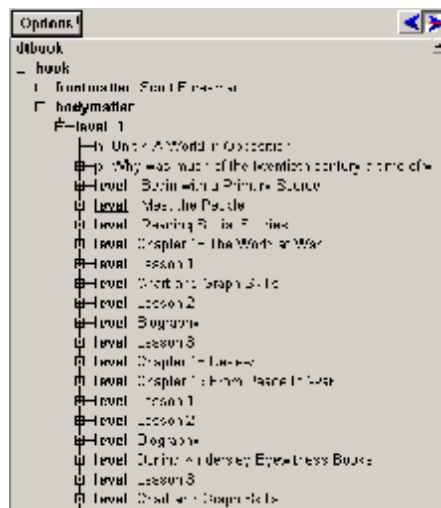
If you click on a tag name in the hierarchy while holding down the Ctrl key, you toggle highlighting of that element, in both panels.

If you drag the mouse, vertically, across multiple elements in the hierarchy, you can highlight elements and their corresponding text. The screenshot below was taken after clicking some  points to open some elements and then two elements were selected, either by clicking with Ctrl or by dragging the mouse across them. The result is highlighting in both panels (see screen-shot below).



Browse the print copy

With a new textbook project, you may want to flip through the print copy while noticing how the publisher has arranged the NIMAS elements. You can switch to the Print View and work your way through the file, on screen, as you flip through the print copy of the book. This is a good way to get a feel for tag semantics. If you can't figure out the way a tag is used, you can consult the brief tag name definitions at the end of this document. Many of the tag names are reasonably mnemonic. And besides, you don't care what they mean, only how they seem to be used, relative to format changes you plan to make.



Reveal auxiliary element information

The "Options" button at the top of the dtbook panel presents a popup menu with various panel control settings. One of those is "Reveal". The initial setting is "Tags only". But you can add a variety of additional information. For example, select "Text" and you get the display at the left.

Now you can see the leading prose for each element, as shown at the left.

Tag names (also called element names) are always shown in bold. Additional data, as selected by the Reveal options, are shown in regular-weight text.

The table below lists the information you can display in the Dtbook panel, and what you might do with it.

“Reveal” settings (multiple choices may be active simultaneously)

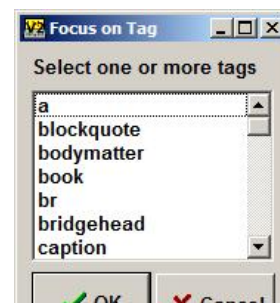
Tags Only	Just the element tagnames show
Depth nr	How deep (how far to the right) the node is in the hierarchy relative to the leftmost "level" node. Depth number is useful for selecting headings of a particular level.
Recursion nr	<p>How many times the same kind of element occurs within itself, e.g., a list within a list. An element with the same tagname does not have to be immediately contained in another element—there may be intermediate elements of some other name, such as a list item "li" that is contained in a list but has another list within it—the list element is still said to have recurred, even when the sequence is list, li, list.</p> <p>Elements that do not recur receive a recursion number of zero (you could argue that they in fact recur once, but using zero gives an inspired conceptual fit to braille format concepts for multi-level things like exercises and lists).</p> <p>Elements that recur two more times are numbered, by tagname, from the top of the NIMAS hierarchy outward. The recurring element nearest dtroot is numbered 1 and the other recurrences receive accordingly higher numbers.</p> <p>A publisher may describe exercise material as a "list". If the list element has a recursion number of zero, it is like a single-level exercise. If the list element has a recursion number of 1, 2, or 3, it is like a multiple level list, with the appropriate indent.</p>
Style data	The braille style name or indent-runover characterization of the paragraph at the given node.
Text	The print text at the given node. Useful when relating Dtbook panel entries to the print copy of the book.
Attributes	The publisher's element attributes. Xml allows each element to have optional attributes. They are occasionally useful.
Page and line	The location where the associated text starts in the file, given as pagenumber.linenummer. The page number is the page ordinal number (as shows in the cursor position readout in the Tools panel).
Block size	The size, in cells, lines, or pages, of the text in the given element in the file.
Braille page nr	The braille page number where element text starts. Useful for navigation and when deciding where to break the book into volumes.
Print page nr	The print page number where element text starts. Useful for navigation and when deciding where to break the book into volumes, and to relate the node to the print copy of the book.

The "NIMAS Touch"™

The "NIMAS Touch" is a trademark of Computer Application Specialties Company, and refers to the special way Braille2000 uses NIMAS tag information (i.e., the dtbook Panel) for summarization, navigation, and control.

Summarization

Using the Reveal and/or Focus options, you can learn about the textbook as a whole, to help plan overall formatting decisions. For example, if you Options and Focus and Focus By Name you get a selection box that lists all the



element names:

You could select all of the headings (H1, H2, ...) which would let you see just the headings in the dtbook panel (the other elements still exist but are hidden; some structure elements such as "level" also show so that the heading elements have places to exist in the hierarchy). You can use Options and Reveal and Text to see the headings themselves. In this way you can get an overall understanding of the use of headings.

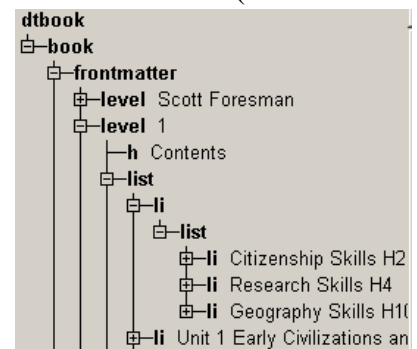
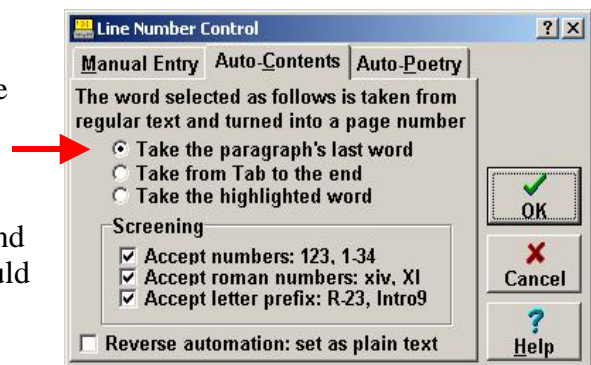
Navigation

By clicking in the dtbook panel you can set the cursor to the corresponding spot in the prose, and vice-a-versa, under the control of the "cursor transfer" buttons. You can use the Show and Focus settings to configure the dtbook panel and then by clicking there you can jump to that part of the prose in the editing panel and vice-a-versa.

Control

By using the dtbook panel to form various multi-highlight selections, you can then carry out global operations on selected prose elements (such as change format, change emphasis, handling contents entries, guide words, diacritics, etc.).

For example, you could right-click on one H4 node and click "Select similar" and "Same tagname". This would select all H4 nodes in the entire document along with all of the associated prose for those elements. You could then right-click and use "Change style" and then select a new style: that style is applied to all selected locations throughout the document (all at once). You could also use the Style button or the style dialog box (via right-click to the Style button). The general theme is "select and act" and it is the most compelling when the select process selects multiple locations in some automated way such as "select similar, same tagname" as used here.



Processing Contents

One area where significant variations are possible is in the markup for the table of contents. A publisher may put the page numbers in an element by themselves (e.g., "linenum") or the numbers may be plain text in a paragraph ("p" element) or a list ("li" element). Our sample textbook describes the contents as a list (actually a multi-level list, see Formatting Lists, below) in which the page number is not set off in any way, other than being at the end of each entry.

To format the contents, you need to find the contents (use the Dtbook panel), handle the entries to get page numbers and guide dots arranged properly, and then to manage the overall format (e.g., as a multi-level list). In this section we deal with parts one and two.

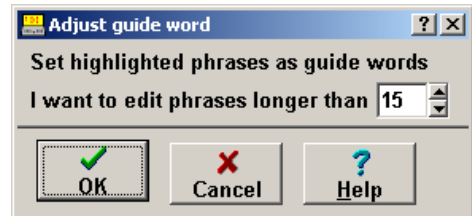
Display the Dtbook panel. You know contents will be an element in "frontmatter". Use Options, Reveal, Text to show the text, and then look for "Contents". You will find it is the second "level 1" element in "frontmatter". Some publishers may identify contents with the xml optional attribute **class="contents"** (one of the rare occasions when the element's attribute may be

interesting). The following formatting steps do not require relabeling any elements but you can, if you want, change the tag to, say, "contents".

Ctrl-click the node that represents the entire contents. This may be a "level" node or you may have changed its tag to "contents" via the above (it makes no difference). Do not click the "h" node, that is merely the heading line. Now the entire contents is highlighted. Click Menu, Do, Line Number to bring up the Line Number Control dialog box. Click the tab "Auto-Contents". For the contents arrangement in our sample textbook, the default settings are good. Click OK. The contents-entry line number mechanism is then applied to everything highlighted and all of the page numbers become contents line numbers, with guide dots.

Set guide word text

A good NIMAS file will use the tag name "dt" to denote each guide word or phrase. You can select all of those via the Dtbook panel and then use "Set guide word text" (in the Do menu) to mark all selected items as tagged guide words.



When the "dt" element comprises a long phrase, it obviously won't fit in the guide-text footer (particularly when there are two entries in the footer). In the dialog box, the "I want to edit phrases longer than xx." can be used to "catch" those guide word entries that are too long. For each one you will be shown a dialog box in which you can arrange the abbreviated guide text for that entry. This can be very tedious, but automation of this task would not likely do a good job (If you don't shorten the entry, Braille2000 will merely truncate the entry to fit the space of the footer line... not very nice.)

Transferring work to V1

While cute and hopefully compelling, V2 is young and it may therefore be temperamental at times. Yet work must get done. There are two techniques you can use to push past any impertinent software behavior:

Edit As Plain Text

For tables, tabbed prose, and multilevel lists, you can highlight such a block of material and click Menu, Edit, and "As Plain Text". Doing this will leave the braille unchanged but will remove all tabs and fancy automation, leaving you with plain text that you can then adjust in any way you wish. Note that some layouts (especially columned tables) will lose column-wrap and line-wrap, becoming a sequence of one-line paragraphs. If you use this tool, you should do all automated adjusting first because after you apply Edit As Plain Text the pizzazz will be gone.

Save As Annotated Braille Text (ABT)

You can save your material using SaveAs (Menu, File, SaveAs) and then set the file type to "Annotated Braille Text (.ABT)". This will give you a file you can then work on using Braille2000 V1. But to tone down the new features of V2 (things that V1 can't understand) the entire file will be converted to "plain text" as per the above paragraph. The plain text conversion affects tables, tabbed prose, and multilevel lists. The braille remains unchanged but for those kinds of structures, the pizzazz will be gone when you work on the file using V1.

Note that it is not necessary to convert your files to .ABT format just to send them out to be embossed: Braille2000 V1 is able to open and emboss .B2K files (V1 can't edit them). The .B2K file produced by V2 has the same basic properties as the .ABT file, in that it can be embossed without Braille2000 software when necessary. Production centers now accepting .ABT files for embossing should be able to handle .B2K files using normal procedures.

Select Pages

The ability to create a file (to save or perhaps just to emboss) that consists of certain selected pages is very useful. For example you can select just a page or two to emboss if there is a problem with the embossing of just a page or two. There are also ways to select just pages containing reserved space for a tactile drawing.

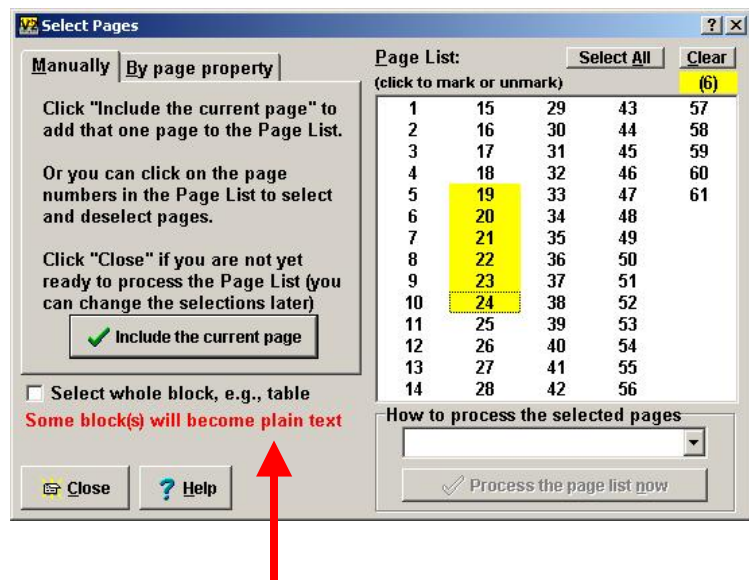
With the advent of tables and nested lists (and later, footnotes and drawings), i.e., those special paragraph ensembles known as "blocks," there are new technical challenges for select pages. For example, a table in column format may span several pages. As a whole it has normal table properties, but if you select just a page that is but a part of the table, it no longer has its regular table properties—it is something else when not whole.

The new Select Pages function uses two strategies to handle blocks (i.e., tables, drawing, nested lists, and other ensembles with gestalt behavior). The first strategy is to automatically expand the page list (i.e., the selected pages) to include all of any included block structure. By this strategy if you select a page that contains part of a multi-page table, the other pages are included so that the entire page is selected.

The second strategy is to automatically apply "Edit As Plain Text" (discussed above) to convert a block that is only partially included to plain text form (the braille is unchanged but any special properties such as table attributes are discarded). As plain text, the part that is on a selected page can be reproduced independently from the rest.

If you select pages that would operate on only a part of a block structure you will see the warning message (the line in red) *unless* you put a checkmark at "Select whole block, e.g., table".

As shown at the left, there are some tables that will become plain text, in the resulting set of pages. If the task is to emboss the pages, this is always OK. If you need to do further editing the loss of block properties (e.g., table properties) will be a handicap.



If you checkmark "Select whole blocks" additional pages will show (in yellow) in the list of selected pages, so that whole blocks (i.e., whole tables) will be selected, with all their properties intact.

Unified English Braille (UEB)

The International Council on English Braille (ICEB) has a braille code known as Unified English Braille. The specification is not managed by BANA, they have adopted it as-is, claiming, among other things, that it is paramount for English braille be the same throughout all English-speaking countries. UEB does not honor Nemeth nor the BANA foreign language codes. At the moment BANA has not issued a statement on the fate of foreign language or Nemeth, saying that those codes are for the moment unchanged and still in use. The announced end-of-EBAE date is January 4, 2016.

UEB in one glance

1. Pay no attention to semantics (same braille for "do" and "do" the first step on a musical scale)
2. No exceptions (use letter signs before letters of enumeration in a list)
3. New names for concepts (its not the letter sign it's the grade 1 indicator, etc.)
4. New (and clever) ways to manage emphasis and displayed material (more below)
5. No "nasty" contractions for: dd ble com ation ally
6. No omission of spaces in braille for to into by a and for of the with.
7. Different punctuation codes
8. No non-Latin, accent, or print indicators as such

But not so fast... there are some obstacles to success

1. there is no o'clock shortform
2. there is a completely different scheme for emphasis(italic, bold, underline)
3. there are changes in special symbols (Lire, paragraph, section, dollar, asterisk, caesura)
4. there are many new symbols (those in the Appendix in 2011 Formats, and more)
5. the rules about when to use contractions are different (**9% of all words contract differently**)
6. there are no semantic distinctions (do, the musical scale vs. do, the verb)
7. the set of shortform words is **greatly** enlarged
8. the syntax for numbers is different (**multiple number signs are sometimes needed**)
9. **everything relating to math is unlike anything seen before**

You should not trust your instincts until you are UEB-certified and it has been some time since transcribing EBAE. The spelling checker and (bi-directional) translator of Braille2000 have been thoroughly tested and should be very reliable.

Not Yet...

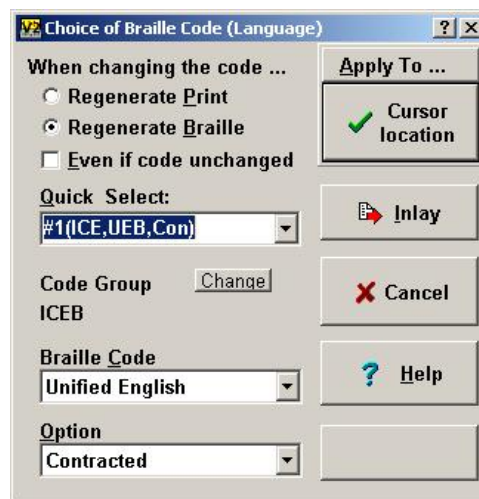
1. At this time, UEB math is partially implemented (more is coming).
2. At this time, the "dot locator for use" is not automated.
3. At this time, who knows whether top boxing lines require the dot locator for use.
4. At this time, who knows whether empty table fields (dash) require the dot locator for use.
5. At this time, who known when 2011 Formats will be republished for UEB (e.g., any changes for guide word footers because of UEB???)

For the reader of general literature (little or no use of emphasis, standard punctuation, no Greek, no special math symbols) the differences can be figured out as you read (most of the contractions are the same, just used with some different contexts). But for the transcriber, there are dozens

and dozens of new special print symbols that will most likely slow you down, and for math... well we won't go there, although the US may soon be the only country using Nemeth.

The important concept to take away is that Braille2000 can be used to generate EBAE from UEB. (Note: UEB will only be offered in V2.)

To change language (in V2), right-click the Language button. The window at the right appears. You will find the BANA-defined codes listed (English, Spanish, French, German, Portuguese, Italian, same as in V1). UEB is not defined by BANA. It is defined by ICEB (the International Council on English Braille), it is an international standard. Click the "Change" button and select the ICEB group, and then you can select Unified English as the code. You can further select the option Contracted on Uncontracted. Who knows when foreign language standards will be updated for UEB. If you want to do a foreign language textbook now, with English portions in UEB, you will have to use one of the ad hoc foreign language codes (Spanish, French, German, Portuguese, Italian) in the ICEB group in order to have compatible punctuation and emphasis indicators. Those codes are completely unofficial.



When you are doing nothing but UEB, you can make it the system default language so it is always pre-selected. To do that select UEB and then click on "Apply To" and choose "System Default" and then click the button (its legend will have changed to "System Default").

This same window can be used to change the braille automatically (i.e., to change UEB into EBAE). To do that, do the following:

Changing a UEB braille file to EBAE (the braille changes; the print stays the same)

1. Open the file of UEB braille.
2. If the Language button does not show Unified English, right-click that button (to get the window above) and set the language to UEB and click the round button for "Regenerate Print". Click Apply To and select All Of File and then click that button. (The braille will remain unchanged; the print back-translation will self-correct to show the UEB in proper print notation.)
3. Right-click the Language button (to get the window above) and select English (you may need to go to the BANA group) and click the round button for "Regenerate Braille". Click Apply To and select All Of File and then click that button. The braille will be replaced with EBAE braille, based on the print which remains unchanged.

You can use the same method to change EBAE to UEB, but remember that EBAE does not back-translate flawlessly, and thus the resulting UEB may be inaccurate in spots (meaning you should proofread it and expect to make some manual adjustments).

UEB Emphasis (A whole new world)

UEB has four kinds of emphasis: italic, bold, underlined, and script. (script applies to decorative

fonts used for special math letter symbols, it is conceptually different from the others, and not well thought out, in my opinion). Note that UEB does not support double-underlined as mentioned in the 2011 Formats manual (there is a provision for transcriber-defined special emphasis, but it consumes extra braille cells).

The concepts of italics, bold, underlined, script, capital, and grade-1 all exist in UEB as applied to single print symbols, to words, and to phrases, and each indicator in different and when being terminated early each concept has its own special two-cell termination indicator.

Furthermore, you can use symbol bold or symbol capital (or any concept) on a contraction and it affects only the first print character. Yes, you can make just the "i" in "ing" bold and still use the "ing" contraction. Contractions cease to be used only when in indicator has to appear in the middle of the contraction, such as in "*child*".

The normal approach is to use a symbol indicator for a single print letter, to use word indicators for whole words up to two in number, and to use the phrase indicator with closing termination indicator for three or more words. Unlike in EBAE the terminator goes right where the effect stops, however you **can** terminate a phrase effect with a word effect or even a symbol effect, but those uses are atypical. You can use the word indicator mid-word and its effect continues to the end of the word. You can use the word indicator in a word and the termination indicator to stop the effect before the end of the word. You can be very creative. I believe it is bad form to use a phrase indicator mid-word.

As in EBAE, the normal usage for multiple adjacent paragraphs with the same emphasis is to re-introduce the emphasis at the start of each paragraph with the termination indicator only at the end of the final paragraph. In theory the phrase indicator at the start does carry through multiple paragraphs by itself, but it is considered cumbersome to the reader to track large blocks of text from the same indicator, so it is repeated for convenience. The exception is for displayed material.

A technique that is unique to UEB is embedding displayed material between a line containing only beginning phrase indicator(s) and a final line containing only termination indicators, as illustrated figuratively below. In this approach (which normally does not span multiple braille pages), the beginning indicators are not repeated on each paragraph. (2011 Formats would also want boxing lines, most of the time, for displayed material.)

Displayed Material Example (from the ICEB UEB manual, the concept being a grid of capitalized single letters)

code for phrase caps and phrase G1	" "	= ; ; ; , , ,
A B C D E F	A	B C D E F
G H I J K L	G	H I J K L
M N O P Q R	M	N O P Q R
terminate G1 code and terminate caps code	" "	= , ' ; '

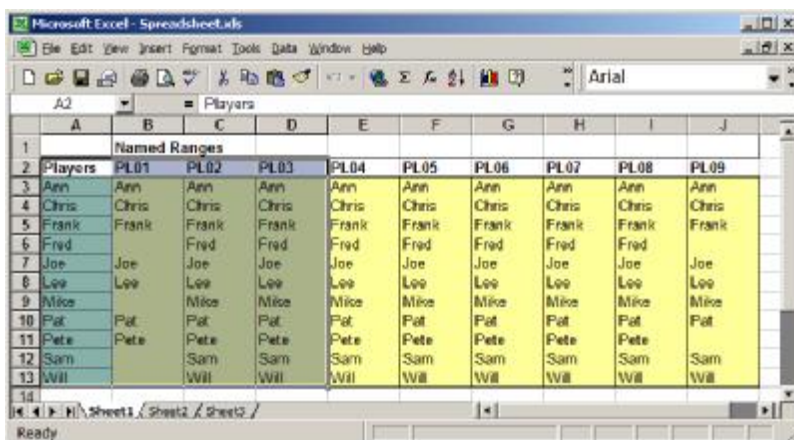
State and emphasis indicators

	Cap	G1	Number	Italic	Bold	Underline	Script*
Symbol	,	;		. 2	^ 2	_ 2	@ 2
Word	, ,	; ;	#	. 1	^ 1	_ 1	@ 1
Phrase	, , ,	; ; ;	# # ∴	. 7	^ 7	_ 7	@ 7
Terminate	, '	; '	# '	. '	^ '	_ '	@ '

*Script denotes the fancy-font letters used in math (such as \mathfrak{R} for the set of rational numbers).

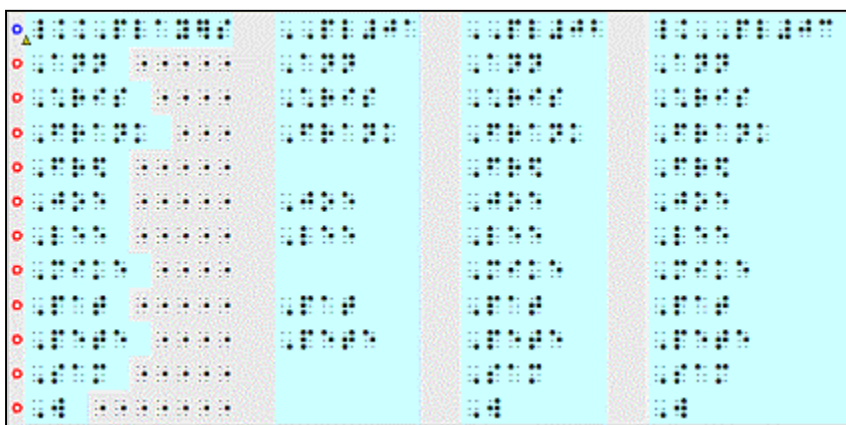
Handling a spreadsheet

Here is a spreadsheet as presented by Excel. Highlight all or a part (here just the first four columns were selected). Copy this to the Clipboard (ctrl-c) and then Paste (ctrl-v) into Braille2000.



The generated braille is shown in the snapshot below.

If the whole spreadsheet were selected, the table would have appeared in staircase format (which you could change to listed or linear, or you could arrange for vertical sectioning).

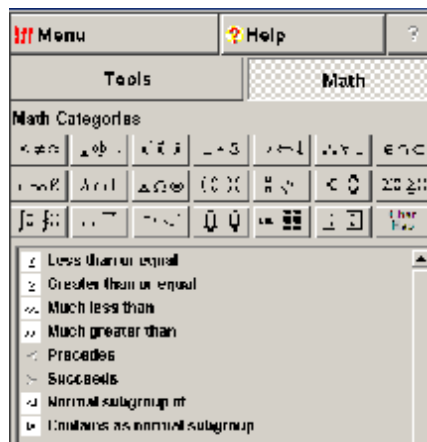


(The underlying simplicity here is that Excel will put a spreadsheet or a portion onto the clipboard as an RTF table that the Braille2000 paste mechanism already knows how to handle!)

Math Toolbar Input

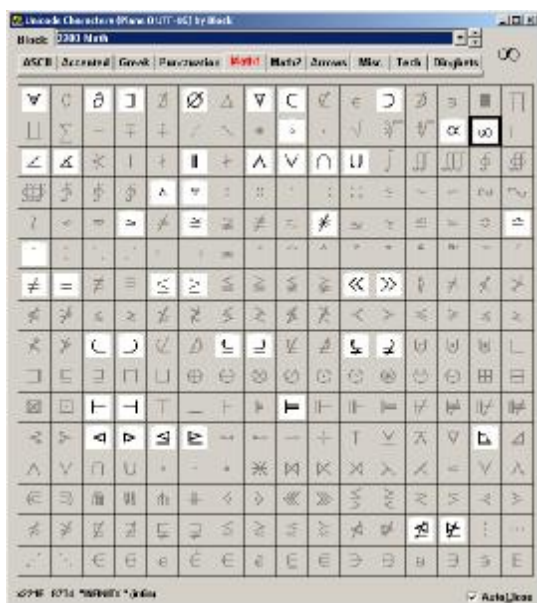
Note: when math toolbar input is more mature, it will be packaged as an optional extra-cost feature for Braille2000 V2. At the present time, it is free.

When the cursor is in a part of the file where the language has math capabilities (i.e., Nemeth, NUBS, or UEB), the Math panel is available to assist with math and special symbol input. (The width of panels is adjustable, so the precise layout may differ a bit from what is shown here.)



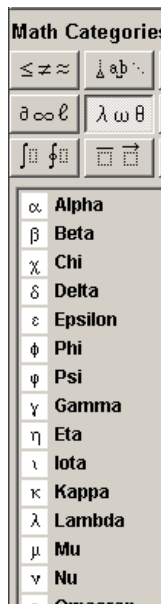
The large buttons select categories of math input items, including symbols, Greek letters, and forms, such as square root and fraction. Not all items are defined by all braille codes, and the implementation of math translation is new to Braille2000 and not all of the Nemeth and UEB math capability is available at this time. More and more capability is coming.

The symbols and forms that are available are shown in white (above, and left). Those in gray are not available (but of course you can use six-key input to do anything).



The char map button (above) displays a comprehensive grid of thousands of symbols (arranged by Unicode sequence, grouped into categories, left) that you can use as input for your document. Symbols available in the selected math code are shown in white and those not available (by translation) are gray. As you can see there are quite a few highly specialized math symbols that the translator does not know or the language (UEB in this case) does not define. The more customary symbols are there.

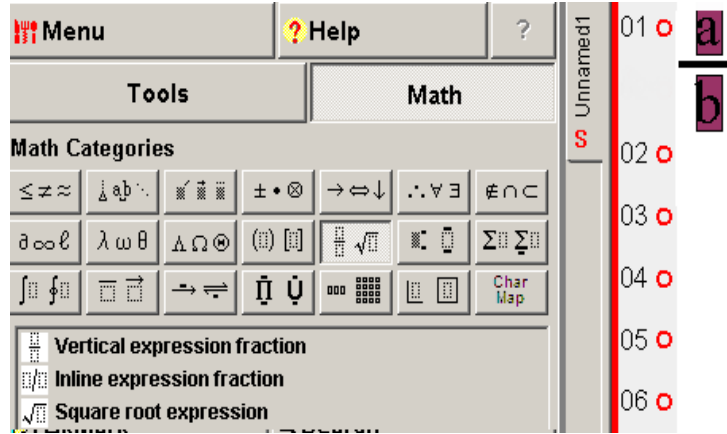
In the Char Map, the tabs select panels with different kinds of symbols and foreign language characters. The choice box at the top (called "Block" identifies the subcategory of the Unicode character set (much of which has no built-in braille translation, but it is still a useful resource for what some unusual symbol is, and its Unicode number (via the readout in the lower left).





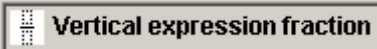
When you click on a white-outlined symbol, it is inserted into your braille file at the cursor.

The more common symbols and Greek letters are handy in the main Math panel, for example, lower-case Greek letters: click the category button to see the list of choices and then click an item (a white item) to have it be inserted at the cursor.

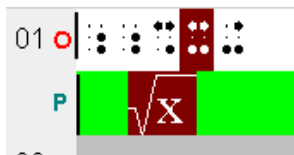
As with filling in tables, after you click a math form, such as a vertical fraction, you get (in print and in braille) that form with placeholder entries (a and b in this case) that are input stops. When you press Tab the "a" becomes highlighted and you can type over it or input a subordinate form (say square root or a variable with an exponent). When you press Tab again, the "b" is highlighted and you complete it in the same way.



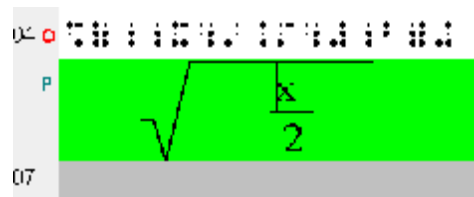
Composition is like building a table...

Suppose you want the square root of x/2. Move the cursor to where the formula should go. Click the Math button and then the  button and then click . You will get the square root of X where X is an input stop (like a blank table field). Press TAB to highlight the X portion and then click . You will then have the square root of a/b with input stops for a and for b. Press TAB enter x (it replaces a). Press TAB, enter 2 (it replaces b) and you are left with the formula you want.

Step 1: the outer form:



Finally, all input stops have been replaced:



Provisional Guidance

It is because of BANA that UEB has replaced EBAE as the operational braille code in the United States. Note that BANA has no true "authority" and merely suggests practices. By "Provisional Guidance" (see their guidelines on their web site) BANA describes a bizarre approach for a math transcription by which narrative text is in UEB code while individual formulas and math expressions are in *uncontracted Nemeth* (never before heard of) with each such expression surrounded by three-cell begin/end "code switch" indicators. The result is considered by everyone as being hard to read and confusing. Nobody outside BANA itself knows what "Provisional" refers to, but it is widely known that (a) BANA operatives in Canada have eliminated Nemeth code altogether and (b) the Library of Congress National Library Service has

requested bids on comprehensive training materials on UEB math code. Both of these moves suggest that "Provisional" means "until current Nemeth readers are exhausted" (perhaps six years) and then Nemeth code will be eliminated. Please note that Nemeth code is considered essential to the study of STEM field subjects, principally advanced mathematics. It is widely understood that UEB math code is not now and never will be suitable for such endeavors (it will discourage STEM field studies in blind students).

For Braille2000 to handle "Provisional Guidance", the document must switch back and forth from UEB to UN (what do we call non-traditional Nemeth in uncontracted form? we will use UN in this writeup, the setting in Braille2000 is "Provisional Guidance") many times per page, and perhaps even more than once on the same line(!). The concept for the transcription is that the main text is in one code (UEB) while other code (UN) peppers the material, like a silver inlay (that is unfortunately tarnished; Braille2000 finds this approach to braille literacy backwards, mean spirited, and contrary to common sense).



In the Code dialog box, there is a new button called "Inlay". Rather than changing the Code to the given setting, the Inlay button also includes changing back to the original code at the end of the segment. It is used in two ways: (1) you can highlight a segment of material and then use Inlay to set that segment in a new Code (this use is similar to clicking "Highlighted Text"), and (2) with nothing highlighted when you click Inlay a text token (the name of the code) is inserted at the cursor, highlighted, and then the code is changed for that highlighted region. This latter function (for which you can make a Speedbraille macro) is what you will need to transcribe according to Provisional Guidance.

The ICEB group (of codes compatible with UEB) has a setting called "Provisional Guidance" which is the UN code mentioned above (uncontracted Nemeth) plus dynamic generation of the begin and end indicators. When you need to write a math expression (assuming you are doing Provisional Guidance), you put the cursor where the expression is to go and you select Provisional Guidance and click Inlay (or press your Speedbraille trigger key). You will then see

something like this. 

For this example, the word "Consider " was entered and the cursor was just after that when Inlay was done. Notice the begin/end indicators and the Code change annotations. And because "nemeth" is highlighted, that filler will be replaced by what you enter next, i.e., it is ready to receive your UN math expression or to receive generated math from the math toolbar.

Bookmarks

Version 1 had three bookmarks: X, Y, and Z. Version 2 has unlimited bookmarks that you can name as letters/numbers or words or phrases. And Version 2 bookmarks can be grouped into categories (if you wish). The



BkMark button responds to various mouse actions: if you left-click the button, you can jump to any bookmark in the



file. If multiple categories are in use, you jump by category and then by bookmark name. If you hold down the Ctrl key as you left-click, you can both Jump To or Set a known bookmark. Known bookmarks include those used in the document plus "standard" bookmarks you declare per category (see below). Initially the known bookmarks are X, Y, and Z, and you can set them via left-click while holding down the Ctrl key. Finally, a right-click displays the Bookmark dialog box, shown here.

To create a bookmark, type its name into the "New mark name" box and click Define. If the bookmark is known, it will appear in the Marks list and to change its location, select the bookmark in the list and then click Define.

You can create new category names: just enter the category name in the Category box and then enter a bookmark name and click Define. "general" is the default category name. You can remove all bookmarks in a category using the Delete button in the Category box.

If you click the "Standard names" tab, you can type in a list of bookmark names using a comma to separate adjacent names. Click the "Store" button (not show until you click the tab) and those bookmarks will become "known", i.e., they will show up when you hold down the Ctrl key and left-click the BkMark button and Use the Set menu option. The default list of known bookmarks is X,Y,Z but you can change that to whatever you want. And you can have known bookmarks in multiple separate categories, if you wish. Just enter the category name before you enter the list of bookmark names and click Store.

The identification of standard (i.e., known) bookmarks is a global setting that will apply to all files. Declaring standard bookmark names does not cause those names to be used in any file... you still set them like before. It is not necessary to make a bookmark known before you declare it in a document. The only advantage to having known bookmark names is that they will pop up in the menus you see when you use Ctrl and left-click on the BkMark button, as a way to set a known bookmark to the cursor location without having to type in its name.

All about Selections

A "selection" is one or more regions of the file. The highlight you see (for cut/copy/paste) is a selection. The red indications you see with spelling checking is also a selection. Braille2000 also has "named selections" that get saved with the file and whose regions track with editing.

The only selection you can manipulate directly is the Highlight selection. See page 6 for the various ways to set and/or manipulate the Highlight. Indirectly, you can manipulate a selection (Highlight or a named selection) using Adjust / Selection (click the Adjust button). Note that a selection may be composed of multiple isolated regions, as generated by you using the Ctrl key, by the "dtbook" (NIMAS) panel, by Find All, etc. Many operations apply themselves to each selected region of a selection. For example, paragraph style settings apply to all those paragraph that show highlighting.

For certain uses, you may find that the selected regions of a selection are not as broad as you wish. For example, given some selected words (as per Find All), you may want to operate on the entire paragraph at each selected word location. Or having found some element of punctuation

(Find All), you may wish to further select the word that it touches, at each location. Such localized expansions of selected regions can be done using Adjust / Selection.

Selection Adjuster

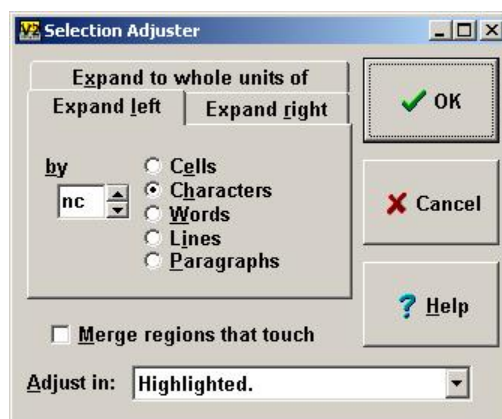
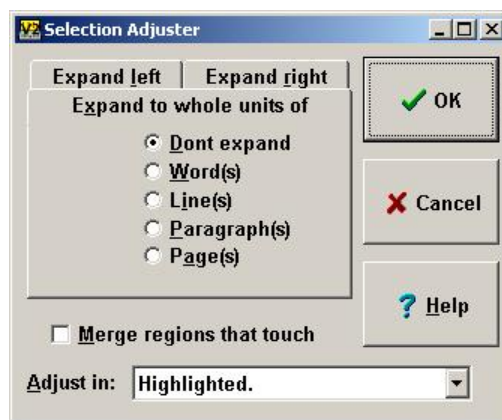
The Selection Adjuster can expand each of the selected regions in a given selection. This operation begins via the Adjust button.

There are three tabs for three composite operations: Expand to whole units, Expand left, and Expand right. In the snapshot you see the Expand to whole units panel. If you select one of the choices (Word, Line, Paragraph, Page) then each selected region has its location and extent "rounded up" to encompass an entire word/line/paragraph/page where it is located. This operation (if used) is done first, before Expand left or Expand right, and if the multiple selected regions then overlap, they are merged together (regardless of whether or not the Merge option is checkmarked). For example, if Find All had selected two words within the same paragraph, and if you used this panel to expand to whole paragraphs, then that entire paragraph would become select (e.g., it would all become highlighted) and what had been two isolated regions would merge into one region.

With or without Expand to whole units, you can use Expand left and/or Expand right to make each selected region bigger by a given number of a given kind of unit. The "nc" at "by" indicates "no change" and when that is showing, nothing will be expanded. Using the up/down arrows you can change "nc" into a number, 0 or more, for the number of additional units to add to each selected regions, either to the left or to the right, depending on which tab is active (Expand left is shown above; Expand right looks just the same).

Expand left/Expand right is performed after Expand to whole units has been done. Setting the number of units to 0 often does nothing but it does expand each selected region to the kind of boundary for the unit type that you select. For example, if you have highlighted the "a" in "because" and set "by" to 0 and also select Words, the left side of that selected region would expand (leftward) to the beginning of the word. If you set "by" to 1, it would expand even further leftward to include also the entire word to the left. You can use the 0 count with Expand left and Expand right to get an effect very much like Expand to whole units.

The "Merge regions that touch" option applies to Expand left and Expand right. If you use Expand to whole units and it causes expanded selected regions to overlap, they are always merged and that merging takes place before Expand left/Expand right operate. Then the Merge option is not active, Expand left/Expand right retain the same number of regions and will expand each such region only as much as could be done without overlapping any adjacent region, i.e., things may not expand as much as requested. Then Merge is active, each region expands as requested and if they then overlap, overlapping regions are joined into a single region (i.e., the number of regions in the selection may decrease).



If the file hold named selections, you can choose a named selection via "Adjust in" and then the operations apply to that named selection.

If there is no highlight at all when you use the Selection Adjuster, it applies itself to an imaginary selection having one region at the location of the cursor. For example if you set Expand left and Expand right to expand by one cell (both set that way), then the Selection Adjuster would create a two-cell highlight centered on the cursor position.

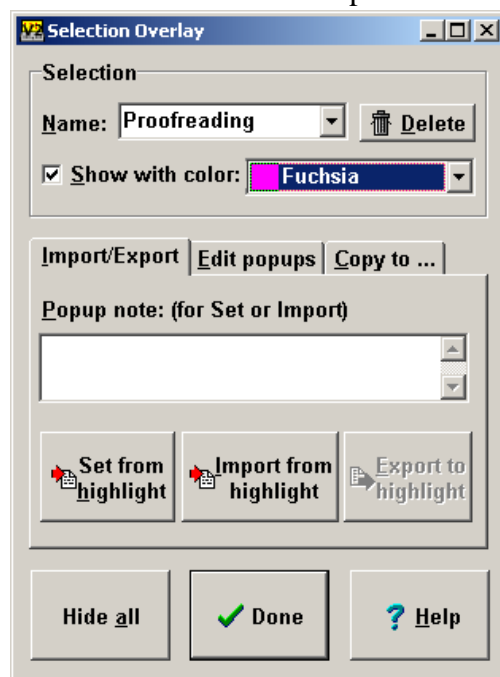
Named Selections (Selection Overlay)

Braille2000 allows for optional named selections that are stored in the .B2K file. (To store a named selection you must be working with the .B2K file type.) To manage named selections, click View and choose Selections.

You do not create a named selection directly. You arrange highlighting and then use "Set from highlight" or "Import from highlight" do define a named selection. You can have multiple named selections, each with a unique name in the context of a single file.

Each named selection can show on the screen (or not show) with a color of your choice. By choosing a color that is distinct from the highlight color, you can see what is highlighting and what is from a named selection. Having a named selection show does not interfere with normal editing or normal uses of highlighting (hence the name "Overlay"). You can change the display mode for each named selection, letting then show (with a given color) or not show. Named selection that are not showing still exist and can be shown later.

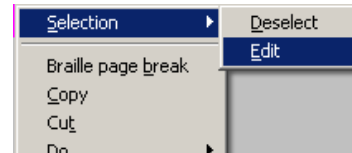
In the snapshot shown above, a named selection called Proofreading is being defined. You might want such a selection if you are doing proofreading for someone else. You can highlight a bunch of words or regions that are problems, and then use the "Import from highlight" button to transfer those selected regions into the Proofreading selection. You can import as often as you wish, building up the Proofreading selection to include all the spots in the file the author should address. If you use "Show with color", then you will see (at all times, until you turn off the Show option) the elements of the Proofreading selection on the screen.



The button "Set from highlight" first empties the selection and then imports from the highlight. It thus makes the named selection identical to what is then highlighted.

The button "Export to highlight" transfers the named selection's regions to the regular highlight, whereupon it can be used with Find or Styles or in other ways.

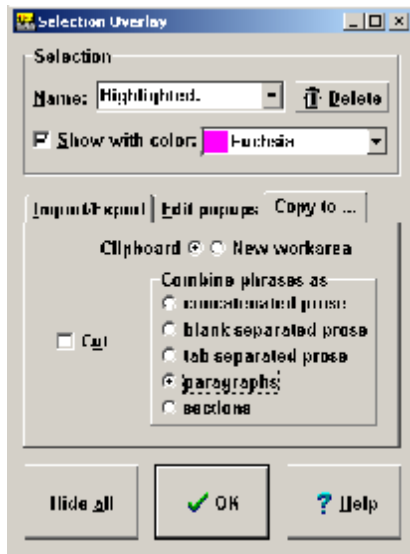
When a named selection is showing on the screen, you may point the mouse at any one of its selected regions and right-click to see this sub-menu:



If you choose Deselect, that element of the name selection will cease to exist. If you choose Edit you get the Selection Overlay dialog box with the Edit Popups tab active and the panel showing the text associated with the selection element you right-clicked on (the text is initially empty). The snapshot shows that the proofreader has entered a note "Missing G1 indicator" to be applied to selection interval #1 (the one that was right-clicked).



Each region (element) of a named selection may have an optional popup caption that is displayed to the reader when (a) the named selection is showing and (b) the mouse is hovered over the selection (hovering means to put the mouse over the selected region and hold it stationary for a few seconds; do not click any mouse buttons).



In this way a named selection can annotate the file in any way you wish, and those selections and annotations are part of the file. If you save the file with such a named selection showing, it will be showing each time the file is opened (until the showing attribute is turned off or the named selection is totally deleted). The task of proofreading someone else's work is an obvious use for this feature, but you can define notes to yourself also, and when you export a named selection to the Highlight, you can do things that operate on each element.

The "Copy to" tab gives you a panel of text harvesting operations. Using it, you can send all of the selected text (that selected by the named selection) to the Clipboard or to a New Workarea in a variety of ways:

concatenates prose: all selected text is joined into one long paragraph without intervening spaces or other material.

blank-separated prose: all selected text is joined into one long paragraph with blanks between each element

tab-separated prose: all selected text is joined into one long paragraph with tabs separating elements (you can use Ctrl+t to show the tab ruler to position such text)

paragraphs: each element of the selection becomes its own paragraph

sections: each element of the selection becomes its own section (retaining certain page attributes such as page numbers)

If "Cut" is not active, the selected prose remains in the file. If "Cut" is active, the selected prose is removed and after the Copy operation is done the named selection becomes empty.

Quick Code Selection

Setting the braille code for a transcription is often done just once per volume of braille, or not at all if the default code is being used. But when doing foreign language or math (in the embedded Nemeth "Provisional Guidance" approach), code changes may be frequent. To assist in frequent code changes, the system organizes code choices in several ways:

1. **The Default Code.** When you begin a new file, the code is preset to the system default code. You can change this code using the choice box at the right (right-click the Code button to get this dialog box). Select the code and then click "Apply To" and select "System Default".

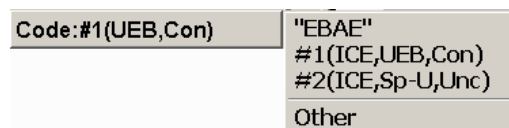
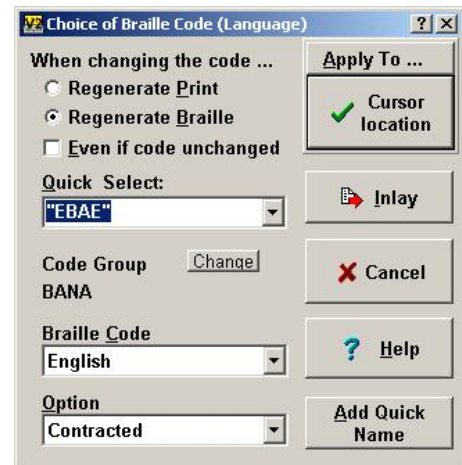
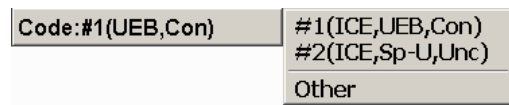
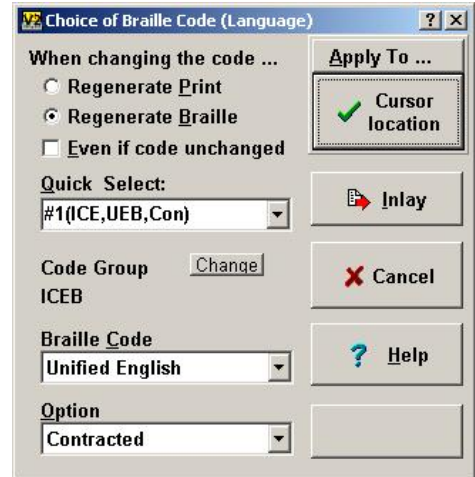
2. **Code Families.** The code dialog box is organized by families (generally by the organization that is responsible for setting the code rules). There are three families: ABL, BANA, and ICEB. For most transcriptions involving more than a single code, the codes will be from the same family. To change the family, click the "Change" button. Then select the code by name and then select from any options that are offered.

3. **Codes Used in the Current File.** A given braille file may need to use a few different codes, say Spanish and English, or UEB and Nemeth. When you click the Code button (shown at the left), you will see a menu of the codes already used in your file. To switch back and forth from code to code, just click the Code button and select the code. File codes are numbered in the order used in the file. The identity of the code is abbreviated to save space. Shown above is ICEB, UEB, Contracted and ICEB, Spanish, Uncontracted.

4. **"Quick" Code Names.** You can define one-word names for codes you use frequently (the name may contain hyphens and underscores but not spaces or punctuation). Each name is a synonym for the code that was selected when the name was defined. Quick names appear first in the menu of Codes (the menu shown in #3 above), and in the Quick Select list (in the Code dialog box).

To define a Quick name, right-click the Code button to display the dialog box (right), select the Code in the usual ways, then click in the text box at Quick Select and type in the Quick name (it will be enclosed by quote marks). Then click "Add Quick Name".

For example, if you do mostly UEB you would have your default Code set to UEB (in the ICEB group). If you occasionally want to work in EBAE, you need to switch groups and although that is not difficult, it takes extra time. The dialog box is shown ready to



define "EBAE" as a Quick name. Once this is done, "EBAE" will appear in the menu choices immediately when you left-click the Code button (whether or not EBAE is a code previously used in the file). Quick names are system-level settings: all of the Quick names you define will be available in all of your files.

To remove a Quick Name, display the Code dialog box and select the name using the dropdown arrow button on the Quick Select control. Then click "Remove Name" (this function is handled by the same button that says "Add Quick Name" in other contexts).

Modified Codes

Certain settings (notably the choice of active dictionaries, via the Spelling dialog box) act as attributes on a given code definition. When you set a code attribute (e.g., via Spelling) to a special value, the definition is "modified" from the system standard. For such codes, the word "modified" will show in the Code Choice dialog box, to the right of the words "Braille Code," and legend of the Code button will include the term "mod".

Grade Relaxer

Carried over from V1, there is the re-translation mechanism known as the "Grade Relaxer". This tool takes a contracted document and selectively uncontracts ("relaxes", i.e., spells out) the level of contraction (the "grade" of the braille) of selected elements. This produces a document tailored to the reading skills of a particular child who is learning contracted braille. This process works on both EBAE and UEB documents.



The main dialog box is shown at the right. You select a profile name (usually named for a particular child) from the list of profiles you have already defined. You adjust the "Text to be relaxed" and select any "Options" and click OK. The relaxer can also change a single-spaced document into double-spaced (controlled either by the Doublespace checkbox or by the "Normal single spacing" profile element (see below).



The default behavior is "Create new workarea", i.e., the original file is unchanged and the relaxed copy is written to a new, unnamed workarea, that you can immediately emboss or, if you wish, Save to a new file. If you select "Y/N each change" then the system stops at each contraction it plans to spell out and asks you (Y/N) for permission for that individual item. If you turn on "Report usage" then a report document (print document) is generated listing each contraction category and the number of times a contraction instance was spelled out.



To create a new profile or further adjust an existing one) you use the tool at the left. It is displayed via the "Adjust profile" button (above dialog box) or by clicking Adjust and selecting "Grade Relaxer Profile". For a new profile, type the name into the Profile name box. For an existing profile, click the dropdown arrow button and select the name you want.

There are 222 contractions listed (including those in EBAE that are not present in UEB, some are listed more than once for different contexts, such as "whole word" vs. "part word"). The list is structured as a hierarchy for ease in navigation. Subordinate entries are shown by clicking a  box or hidden by clicking a  box. Clicking a color-dot applies the Click Action to it (toggling between green and red as the default action). Clicking a category dot changes all subordinate elements to the new setting. Green denotes those contraction types the child can read (those not to be relaxed). Red denotes those types the child has not learned to read (those instances will be spelled out). The "Gen Report" button creates a hard-copy print listing of the profile.

A green dot for "Normal single spacing" means that normal spacing is OK. A red dot there causes the text to become double-spaced as it is relaxed.

Analyzing a document with the Grade Relaxer

You might be interested in knowing the frequency of usage of the various contractions in your document (say for a volume of a textbook). You can get that analysis using the Grade Relaxer if you do the following: prepare a profile for a child who reads no contractions at all (you might call it "G1"... it will spell out everything). Then apply that profile to your document and click the checkbox for "Report usage". Let the Grade Relaxer create a new workarea (that you will then discard). The generated report will tell you how many instances exist in your document, for each contraction type.

Printout Color Control

Screen colors are not necessarily right for printouts. Each kind of printing (the Print As choice) has its own Options reached through the Options button. Within options (just above the page margin settings) you will find the Custom Colors button and you can use it to get the Printout Colors dialog box (right).



Braille2000 stores two color schemes for printing: monochrome and color. Only one scheme is active any a time (and Braille2000 cannot tell whether you are printing in monochrome on a color printer... you need to select the color scheme that is appropriate).

To set printout colors you should: (1) select the color scheme via "Color choices for"; then (2) select an item (one at a time) that you wish to change; then (3) select the type of color for that item via "Color is"; and (4) choose the color you want that item to print in (a drop-down list of color choices will be available to the right of the selected item).

In the "Color is" selector, "Display color" causes the selected item to be colored like that same item on the screen (the color entry will be that of the screen color). If you select "Schema choice" then the choices for the selected item will be the names of Windows standard colors from your Windows desktop definition.

The color settings are common to all printout options (to all choices of Print As).

INDEX

A

ambig trans
 explanation of, 11
annotations
 change from inception, 14
 change starting here, 14
 changing use of, 13
 control of, 12
 control of visual appearance, 4
 dialogue box, 13
 explanation of, 4, 12
 hover for description, 12
 inspect function of, 13
 moving with cursor, 13
 not used for document control, 12
 remove, 14
 show/not show, 12

B

blank page
 inserting, 17
blue blocks, 2
bookmark function, 14
bookmarks
 explanation of, 42
 instructions, 43
boxing lines, 15
 types of, 15

C

clipboard, 8
code dialogue box
 inlay explanation, 42
 math, 42
code selection, 47
 default, 47
 families, 47
 naming, 47
 used in current file, 47
color scheme
 default not light cyan, 3
 gray, 3
 light blue, 3
 red, 3
 uses of, 3
 white, 3
 yellow. *See*
colored backgrounds. *See* color name blocks
compatibility of version 2
 with Windows 8 and 10, 1
control key
 and arrow key behavior, 4
control panel. *See also* specific buttons
 button explanations, 14

 explanation of, 14
 hiding. *See*
 NIMAS files, 14
copy. *See* cut, copy, paste
Ctrl key
 actions, 9
 braille actions, 10
cursor
 home and end key behavior, 4
 inter-cell gap, 4
 movement of positions, 4
 pg up and pg down behavior, 5
 two positions between cells, 2
 within yellow and gray areas, 3
cut, copy, paste, 8
 change cursor location, 9
 cursor location, 9
 paragraphs, 9
 phrases, 9

D

demo mode
 non-use of, 2
display modes
 Ascii view, 3
 braille view, 3
 explanation of, 3
 print view, 3
 source view, 3
do functions, 14
drawing spaces
 explanation of, 17

E

EBAE
 changing to UEB, 37
editing
 backspace key, 6
 changing paragraphs, 16
 delete key, 6
 enter key, 5
 highlighted material, 17
 keys used, 5
 symbol key, 5
 tab key, 6
 using the keyboard, 16
emphasis. *See* typeforms
English Braille American Edition. *See* EBAE
explanation of version 2, 1

F

find

- explanation of, 11
- formatting layout
 - centering, 23
 - font choice, 24
 - indent and paragraph, 23
 - instructions for directives, 24
 - italics versus bold, 23
 - list of percent options, 24
 - table of braille to text equivalents, 23
 - tabs, 23
 - translation typing directives, 24
 - unique in braille, 25
 - with translation typing directives, 24

G

- grade relaxer
 - contraction selection, 49
 - default behavior, changing, 48
 - editing, 48
 - explanation of, 48
 - profiles, 48
- green line. *See* interpreter line

H

- highlighting
 - block with keyboard, 8
 - block with mouse, 7
 - changing emphasis, 8
 - coverage of, 6
 - explanation of, 6
 - margins, 6
 - multiple phrases with keyboard, 8
 - multiple phrases with mouse, 7
 - named and stored selections, 7
 - paragraphs, 6
 - paragraphs circles, 8
 - persistent, 7
 - with keyboard, 7
 - with mouse, 7
 - with mouse clicks, explanation of, 8
 - with mouse clicks, list of, 8
- hover
 - explanation of, 12

I

- ICEB
 - explanation of, 36
- import files
 - explanation of, 14
 - rtf into version 2, 2, 27
 - via paste option, 27
- information card, 7
- insert function, 14
 - explanation of, 11
- installation
 - instructions for, 2
- International Council on English Braille. *See* ICEB
- internet license, 1

- interpreter line
 - explanation of, 20
 - multiple display options, 20
 - options for display, 20
 - setting for, 20
 - with foreign languages, 20

K

- keyboard
 - left and right arrows, 4
 - navigation with, 3
 - shortcuts, 9
 - up and down arrow keys, 3

L

- license transfer, 2

M

- margins
 - adjusted, support for, 2
 - use of arrows with, 4
- markup panel
 - auxiliary options, 31
 - changing the width of, 22
 - element tagnames list, 32
 - explanation of, 22
 - focus mode, 30
 - global operations, 33
 - guide word format, 34
 - highlighting elements, 30
 - navigation, 33
 - navigation of, 22
 - navigation using NIMAS tags, 30
 - NIMAS, 30
 - print view switch, 31
 - summerization, 22
 - table of contents format, 33
 - two panel display, 29
 - viewing options, 29
 - whole textbook view, 32
 - with Ctrl key, 22
- math
 - available symbols, 40
 - braille code, 1
 - character map, 40
 - character map instructions, 40
 - provisional guidance, 41
 - provisional guidance dialogue box, 42
 - toolbar description, 40

N

- named selections
 - assigning color to, 45
 - deselect, 46
 - explanation, 45
 - how to set, 45
 - optional popup, 46

- selections operations, 46
 - uses of, 45
- Nemeth code. *See* math: braille code
- nested lists
 - creation of in version 2, 10
 - explanation of, 10
- new features, 2
- NIMAS. *See* markup panel: NIMAS

P

- page break, 17
 - braille, 2
- page control
 - explanation of, 17
- page view
 - scrolling, 5
- paragraphs
 - explanation of colored circles, 3
 - marks, 9
 - shown by use of colored circle, 3
- paste. *See* cut, copy, paste
- percent sign. *See* translation typing directives
 - indicator for translation typing directives, 25
- printing
 - choosing colors, 49
 - display colors, 49
 - setting colors, 49
- provisional guidance
 - for math, 41
- pv button, 5

R

- replace
 - explanation of, 11
- rtf files, 21
 - braille document output, 21
 - using copy and paste, 21

S

- select pages, 35
 - blocks of materials, 35
 - using edit as plain text, 35
- selection overlay. *See* named selections
- selections
 - adjuster dialogue box, 44
 - explanation of, 43
- source files
 - explanation of, 17
 - rtf files, 18
 - toolbars for, 18
 - types of, 17
 - xml files, 17
- special paste
 - options, 29
 - preset choices, 29
- speedbraille keys
 - copy table to clipboard, 19
 - explanation of, 18

- not compatible with version 1, 18
- paragraph styles, 20
- reassign, 20
- recording, 19
- set availability to file, 19
- set availability to system, 20
- styles, 20
- version 2 differences, 2
- spreadsheets
 - instructions for import, 39

T

- tables, 15
 - columns and rows, 15
 - creation of, 15
 - inserting information into, 16
 - insertion of, 15
 - layout pattern, 16
- text to braille layout. *See* formatting layout
- transferring files between versions
 - edit as plain text, 34
 - save as annotated braille text, 34
 - select pages, 35
- translation typing directives. *See also* formatting layout
 - code naming, 28
 - defaults to contracted, 28
 - explanation of, 23
 - language options, 27
 - list of percent options, 24
 - list of special in braille, 25
 - macro options, 29
 - special in braille, 25
- typeform
 - functions, 14
- typeforms
 - adjusting in source files, 21
 - list of symbols, 38
 - retained from source file, 21

U

- UEB, 36
 - beginning and ending symbols, 38
 - changing into EBAE, 37
 - changing into EBAE automatically, 37
 - changing into EBAE, instructions, 37
 - differences between EBAE and UEB, 36
 - elements not yet decided, 36
 - math, 1
 - new elements, 36
 - setting as default, 37
 - support, 1
 - typeforms, 37
 - typeforms, multiple paragraph, 38
 - typeforms, types of, 38
- Unified English Braille. *See* UEB

V

- version 1

- compatibility, 1
- version 2
 - aspects which don't work, 1
 - beta testing, 1
 - differences, 2
 - explanation of, 1
 - new features, 2

transferring files to version 1, 34, *See also* transferring text between versions

W

- Windows support
 - dropped for 95, 98, and ME, 2