

Braille2000, LLC

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

Textbook Transcriptions via Braille2000 Using Rich Text (RTF) Source Material

How Rich is Rich Text

"Rich" text denotes prose that has attributes beyond what plain ASCII character text has. In Windows systems the "Notepad" editor handles plain character text... there is no provision for italics or bold or layout control (no centering or margins). "Rich" text has those additional aspects, and to specify their manifestation (e.g., emphasis, margins, font, color), control directives must be mixed into the prose as a kind of *markup*. Document markup is today a very standard thing... there are various kinds of markup, in various kinds of files. RTF, DOC, DOCX, B2K, ABT, DXP, HTML are perhaps file types you use and they all have different notions about markup, but in each case there are basically two *channels* of information: the prose itself and the markup that defines the presentation of the prose.

Rich Text is Ubiquitous

Years ago, Microsoft was the main player in *What You See Is What You Get* (WYSIWYG) document preparation, with its Word application for the Macintosh as well as for Windows. Microsoft desired control over its trade secrets and so the internal nature of the DOC file were never published but instead Microsoft defined and did publish details of an alternative format called RTF, i.e., Rich Text. The gambit was not totally successful: third parties succeeded in reverse-engineering DOC (and now also DOCX) file internals and figured out how to process those files directly (something Microsoft had hoped to avoid). Meanwhile, the RTF specification has been well documented and occasionally enhanced in order for every nuance of a Word document to have an equivalent form as RTF data.

Furthermore, because RTF is the intended public specification for prose with document attributes, it is also the data form that essentially all text processing tools (word processors, browsers, spread sheets, e-mail handlers, etc.) put onto the Clipboard when you Copy prose... RTF is thus the best possible way to exchange text between applications. And happily, the RTF data that gets copied to the Clipboard is exactly the same as the data that gets written into an RTF file. That commonality is what makes RTF so useful.

But RTF is Old

In modern times, many tools have gone to using XML notation for markup. XML is a natural notation for

this purpose... its little brother HTML has been used to handle web-page presentation on a browser for many years. RTF data is not at all similar to XML or HTML... it is a quaint notation that works just fine for document processing, but you should not expect it to be adapted to other tasks... you won't be seeing web pages written in RTF! But because it has been around awhile and because RTF goes back to the early days of the Mac and the PC, RTF is still everywhere, either directly or indirectly. And that is why Braille2000 uses RTF as a main import format for source (print) documents.

Standard Document Concepts

The applications known as Word, Wordpad, WordPerfect, and their knock-offs provide a comprehensive set of text-processing tools for arranging prose into a document. The features of RTF have been made sufficient to control this process... everything is there as parameters of markup statements, interspersed with the prose. Fortunately you don't need to see or understand it... the application handles it all in a (usually) intuitive on-screen presentation of the layout achieved.

The central notion of the word processor application (e.g., MS Word) is that the prose of the document is arranged into pages according to the document properties (according to markup). The document is described via markup with various parameters for pages, paragraphs, and passages for concepts such as breaks, margins, fonts, and emphasis. Some of these various parameters (all of which are expressed in the RTF data for the document) are used by Braille2000 to guide the layout of the resulting braille and to aid the print to braille automatic translation process. The considerations mentioned here are applied when a print document in RTF form is imported into Braille2000, whether via the "Open Source File" operation (when opening a RTF file) or via the Paste operation (when importing Clipboard data that is RTF data prepared by a Copy operation in a word processor, browser, or spreadsheet). The parameters and behaviors described below are for the most-used tool, MS Word.

Page Effects

Via "Page Settings", the user can declare the size of paper being used and the page margins. These settings are coordinated with information for the currently selected printer, especially with respect to the sizes of sheets of paper the printer is prepared to handle and the maximum size of the printable area on each page (some printers are mechanically limited as to how close to the edge of the paper they can actually ink). Settings involving distances are often expressed to the user in inches although within the application distances and sizes are often given in *points* (n/72 inch) or *twips* (n/1440 inch), for example 12 points is a sixth of an inch (12 point text has letters that are 1/6 inch tall). The four page margins (top, bottom, left, right) state the widths of a border along the outside edge of each page that does not get inked... the size of the printable area of each sheet is the size of the paper minus the margins. An additional space inside the top and bottom margins is reserved for the page header and footer, respectively. Unlike braille, page numbers and running headings go into reserved areas of the page. Based on the choice of printer and size of paper and margins, the system knows the area that can receive prose on each page and the system checks to see that the margins are large enough to avoid trying to ink anything closer to the edge of the paper than the printer can actually print. (It is possible, when using very small page margins, that choosing a different printer (with different minimum page margins), will force a (small) change in printable area of the page sufficient to change the flow of the prose on the pages of the document, i.e., in some (few) cases, when you send an RTF file to colleague for printing, it might not print exactly the same.)

The Breaks

The user can designate places in the prose that are the boundaries between various layout elements. Every document has at least one section, paragraph, line, word, character. Those are the names used for the various chunks of prose. Except for word and character units, the user may insert a *break* to divide the

prose into multiple sections, paragraphs, and lines. Many documents are all one section. By inserting a section break multiple sections can be defined, each with its own page margins and header and footer sizes and content. (Braille2000 ignores section breaks.)

Prose flows section by section, page by page, paragraph by paragraph... there are as many of each as needed to present the material. A page break is inserted by the user to force the immediately following material to start on a new page. In MS Word, page break is signaled by entering Ctrl+Enter (or via tools in the menu or ribbon). During import, Braille2000 handles page breaks in user-defined ways.

Paragraph breaks are very common and are indicated by the use of the Enter key when inputting prose... after you press Enter the next material starts a new paragraph positioned according to the paragraph's first line indent setting. The usual notion is that the system will automatically subdivide the prose of a paragraph into lines based on the width of the printable area and the paragraph's margin and indent settings (arranging lines is a tedious process, so it is fully automated). During import, each paragraph of print becomes a braille paragraph.

Line breaks are optional and often unnecessary... after a line break, the next material starts on a new line according to the paragraph's left margin setting. In MS Word, a line break is signaled by the use of Shift+Enter. During import Braille2000 marks the resulting place in the braille paragraph and also starts a new line at that spot, placing the braille at the paragraph's runover position.

Word breaks are simply spaces or dashes in the prose (characters called *whitespace*).

(Character break is not a useful concept, as the characters of the document (or *glyphs* on paper) are the atomic elements of the prose... they are non-dividable.)

Managing Presentation: Text Placement

Given that the page is a two dimensional presentation, there are settings to manage both the horizontal and the vertical positioning of the prose. As an aid to the author, the settings are general in nature to support as much automated layout as possible. Via the kinds of settings mentioned here, prose is arranged within the printable area of each page.

Horizontal positioning is handled by per-paragraph attributes known as left margin, right margin, and first line indent (these margins apply within the printable area and are separate from the page margins). The margins indicate how close to the edge of the printable area the prose of each line may go. The first line of each paragraph begins at a position that is the sum of the left margin and the first line indent. The first line indent may be positive, and then the first line begins to the right of the margin (e.g., an indented paragraph), or the first line indent may be negative, and then the first line begins to the left of the margin (e.g., the so-called *hanging* indent). But the paragraph is not allowed to ink outside the printable area of the page, so what actually happens with a hanging indent is that the first line is at "zero" (the beginning of the printable area) and the body of the paragraph is offset the given amount to the right. Lines of the paragraph other than the first one thus begin to the right of the start of the printable area by either the size of the paragraph's left margin (if regular indent), or by the combined sizes of the left margin plus the size of the hanging indent (if a hanging indent is used). Lines end at or before the right side of the printable area minus the amount of the paragraph's right indent.

During import, Braille2000 can use the sizes of the paragraph's left margin, right margin, and first line indent, to infer a similar paragraph shape in terms of braille indent and braille runover. This is done on the (arbitrary) scale of one braille cell (of indent or runover) per 1/8" of print margin or print indent. Many

authors use indented print paragraphs where the first line indent is 0.25" and which moves the braille indent position by two cells, i.e., the scale of conversion works for many (but not all) print layout conventions. The Braille2000 user can disable this braille paragraph style inference process.

The print paragraph also has the property of *alignment* (left, centered, right, justified). During import, Braille2000 notices centered vs. non-centered alignment and performs left alignment or centered alignment in braille for that paragraph. (Braille2000 does not handle right alignment.)

Vertical positioning of prose is usually fully automated: the paragraphs (automatically arranged into lines) flow down each page and continue onto subsequent pages as needed, filling each page's printable area. The user may insert section breaks and page breaks to promote prose to the top of the next page. During import, Braille2000 ignores these breaks unless the user has enabled certain behaviors tied to page breaks. (Section breaks to a new page are handled as page breaks; section breaks that continue on the same page are ignored.)

The flow of prose across page boundaries is often managed in print via the concepts *keep together* and *keep with next*. They are very useful to prevent headings and subheadings from being separated (vertically) from the prose being introduced. When a paragraph has the property *keep together*, then if not all lines of the paragraph will appear on the same page, an implied page break is created just before the paragraph so that it starts a new page where it will be complete on adjacent lines. When a paragraph has the property *keep with next*, then the behavior is similar to what was just described except that the implied page break occurs when the last line of the paragraph and the first line of the following paragraph would fall on different pages. When that happens, the paragraph is moved to the top of the next page where it and the paragraph that follows it will be adjacent. Keep together and keep with next can be used together when desired. During import, Braille2000 applies the same concepts (keep together and keep with next) to the resulting braille paragraphs, and they work the same as in print.

Vertical spacing in print is also influenced by a paragraph's *line spacing* and *space before* and *space after* properties. During import, Braille2000 generally ignores those settings, except that paragraphs whose alignment is centered and that have generous space before and space after settings are imported as centered headings in braille (not just as centered prose). Similarly, paragraphs with an inferred braille indent setting of 5 or 7 and with generous space before are imported as cell-5 or cell-7 subheadings, respectively. (Braille2000 will notice declared space in the print settings and use it to infer the notions of blank line before and/or blank line after, as pertains to braille headings.)

Vertical spacing is sometimes arranged through the use of extra blank lines (blank paragraphs) in print. During import, Braille2000 generally ignores blank lines in print. The user can change this behavior.

Managing Presentation: Text Style

Today's print author can choose between a variety of glyph images via the choice of font and the size of characters in the selected font. Furthermore, several kinds of *emphasis* are available (italics, bold, underlined, strikethru). In UEB terminology, these are changes in typeform. But note that the UEB typeform "script" is not actually an emphasis option for print... it is a change in font face. (See the insert on typography, below.) During import, Braille2000 generally ignores the choice of font but it does import standard emphasis usage (italics, bold, underlined) into the generated braille. There are a few special font face names that get special handling. The special names can be changed by the user. A special font face name is any name that contains any of the following keywords as a part of the face name:

1. Braille. Prose in a font named using this keyword is imported as-is,
2. Courier. Prose in a font named using this keyword is imported as computer braille code (when doing EBAE transcriptions) or in uncontracted form (when doing UEB transcriptions),
3. Script. Prose in a font named using this keyword is imported bearing the UEB script typeform.

During import, Braille2000 ignores the size of font that is used. The color of the printed text is given in the RTF data. During import, Braille2000 ignores font color.

Some documents have special text (fields, hidden material, hyperlinks) mixed with the prose. During import, Braille2000 generally ignores special text. The user may declare a portion of the regular text to be hidden from import as well.

Typography

In ancient writing, scribes copied books by hand. A major breakthrough was moveable type (metal slugs with one character each that were assembled into pages, inked, and then used to print a page at a time). The slugs came in sets with the same graphical conventions as to style and size and built-in emphasis. The slugs in a set were said to have the same *face*. Among the stylistic notions of what the glyphs should look like, we find differences in the *stroke* of the lines of the letters. For example, some formal fonts are ornamented with *serifs* on line endings (little barbs, looks official). Times Roman is a font face with serifs. Other fonts have no such ornamentation and are known as *sans-serif* fonts. Arial and Helvetica are of this style. As with moveable metal type, most fonts are variable pitched (different glyphs have different widths), but since the invention of the typewriter there are fixed pitch (monospaced) fonts such as Courier that mimic typewritten material. Today's computers of course use digital fonts, with a clever representation (Truetype and Opentype) that allows the set of glyph shapes to be dynamically scaled and adjusted to render characters in any size and with standard variations of style for italics and bold. For example, the shape of the letter W can be rendered on a slant to show italics or the stroke width can be thickened to show bold or it can be drawn with an underbar to show underlined. For italics, bold, underlined, and strikethru, the same glyph shape can be drawn in a manner to show it with the given emphasis. But there is no stroke transformation to yield script! A script font is simply a different set of shapes... it is a different font altogether, as would be a Gothic font. The UEB typeform *script* is thus not in the same league as italics and bold and underlined. (This author's guess is that script is a typeform only because that face is occasionally used for standard letter symbols in math and science. Perhaps.)

Import Based on RTF Data (and nothing more)

As summarized above, there is a lot of document markup in every RTF file (or Clipboard data) that can be used when importing print into Braille2000. And that means that you can exploit this behavior by performing strategic edits to the print document before you import it. There are some transcription tools (e.g., Duxbury DBT) that work off of the names of Word styles applied to the prose. Braille2000 does not work that way. Yes, you can apply Word styles to your print document, but it is the effect of those styles (say on paragraph margin and first line indent) that Braille2000 honors, not the name of the style.

Braille2000 will derive the following things from a print document during import. The user can adjust some of this behavior, as described in subsequent sections.

1. paragraph prose units
2. line breaks
3. translation (as suggested by font face name keywords: braille, courier, script, everything else)
4. centered vs. non-centered
5. indent and runover (on the scale of 1/8" per cell for paragraph margins and first line indent)
6. keep together and keep with next
7. paragraph style w.r.t. space above and space below (only for headings)
8. emphasis (italics, bold, underlined)
9. non-breaking spaces

But note what is missing from this list of native information.

1. page numbering
2. headers and footers
3. text boxes
4. typeset math

Braille2000 does not read the prose in page headers and footers. That prose is not actually in the RTF data (although the specification for it is). In particular, page numbers in Word are dynamically determined by the current page layouts and are not actually present in the text (and could be at the top or bottom of a page or not show at all). Braille2000 will read the prose inside text boxes (e.g., marginal notes), but does not notice if it is in a box nor note where it is relative to other prose. Standard Word does not offer typeset math, although there are Word add-ons for that feature. The missing elements can be added to the document by the transcriber, in Braille2000, after the prose has been imported. Prior to import, the transcriber may want to adjust the previously listed nine aspects of the document to better achieve the desired braille format and translation.

Besides taking RTF data "raw" as just discussed, a comprehensive set of open markup directives (things visible to the print editor) are available to handle many more aspects of the import process. The directives are all written with a leading % character and are thus known as *percent codes*. (Percent codes are unique to Braille2000.) The next sections of this document explain how to use a mix of raw prose and percent codes to handle various standard elements of textbook transcription. Note that using percent codes is optional and in lieu of edits that can be done after import right in Braille2000 itself. If you choose to prep the print material with some percent codes, then there should be less editing of the resulting braille... choose what works best for you.

Percent Codes

As has been mentioned, a *percent code* is an unbroken (i.e., no internal spaces) keyword utterance that you insert into the prose of your RTF document. Such codes always begin with a single percent sign (%). This use of the percent sign (at the start of a letters sequence) is essentially unheard of in normal discourse. But if your manuscript does happen to have something like %ABC you can make it into normal prose (a) by doubling the percent sign (as %%ABC what will import as %ABC), or (b) by using the percent code %off that disables all percent code processing (but note that after %off, the percent sign is taken as normal text, and so there is no way, in the same document, to turn special processing back on... there is no %on!).

Each %-code must be set off from adjacent normal prose, by one or more spaces, or by a paragraph break (i.e., by being in a separate paragraph by itself). A few %-codes (e.g., %head) *must* be in a separate paragraph just before the prose being controlled. The keyword part of a %-code is case insensitive, font insensitive, emphasis insensitive, and color insensitive. Some added data (e.g., the page number in %pn=) is taken verbatim so you can designate both roman numeral IV as well as iv.

You can inspect the Percent Codes appendix at the end of this document to learn the behaviors of all the percent code keywords. Most %-codes can be placed adjacent to normal prose or on a line (a separate paragraph) just above normal prose. The exceptions to this are the codes that process a whole paragraph, such as %head, %subhead, and %runhead.

Import processing of document operates in either *implicit* or *explicit* mode. The initial setting (the default setting) is implicit, as if %implicit has been used to start. The %-codes %implicit and %explicit are opposites and can be used to change the working mode within the document, as desired. The difference

between them is most easily explained by saying that in explicit mode, the paragraph margins and first line indent are not used to influence the imported braille paragraph indent and runover settings. Emphasis, centering, and keep notions are still active influencers of braille prose. Without any implied braille paragraph formatting (e.g., explicit mode), all braille paragraphs would usually have a 1-1 (indent-runover) arrangement (probably not what you want). When using %explicit, the transcriber almost always inserts various additional %-codes to set the format of the imported braille paragraphs (such as %indent, %list, etc.). When you do not use %explicit, and also after using %implicit, the print paragraph margins, first line indent, and space before and space after attributes are used to imply imported braille paragraph indent and runover. But there are exceptions to this: after a %-code format directive (e.g., %list) some of the following paragraphs will be imported according to that directive and not according to implied style.

When working in implicit mode, you can override the implied braille paragraph style using %-codes. The setting will apply to the following paragraph and to additional following paragraphs that have the same print margins and first line indent as the print paragraph immediately following the %-code. So if you have a number of print paragraphs in sequence, all with zero margins and first line indent of 0.25", and you put %list1 before the first such print paragraph, the "list1" format (indent at 1, runover at the highest indent in the same list plus two cells, i.e., as per a multi-level list) will apply to all of the adjacent print paragraphs that have the same left margin, right margin, and first line indent. At the first following paragraph that is different in those settings, implied braille paragraph formatting will resume (unless there is a %-code there to override implied format again). (Remember that in explicit mode, via %explicit, implied behavior *never* resumes.) Keeping this in mind, it is perfectly feasible to use a mix of implicit and %-coded format transforms. Remember that Braille2000 itself is a WYSIWYG braille editing tool, and all the effects of %-codes can be done by editing the imported document later in Braille2000 itself.

Concepts for Textbook Transcription

(Note: the concepts discussed here are more completely defined in the publication *Braille Formats Principles of Print-to-Braille Transcription, 2016*. You are referred to that publication for further information.)

Once upon a time, books were transcribed by six-key input from a transcriber working from a print copy. Today, digital data is used, obtained from the publisher or perhaps by scanning and OCR conversion. Sometimes a PDF file is provided, but a PDF file is not a source file (it is difficult to use directly; the PDF file should be processed by an OCR tool to derive DOC, DOCX, or RTF data... the OCR tool will infer document structure (e.g., paragraphing) from the spatial layout of the page images). Some publishers supply NIMAS (XML) data files. Braille2000 can process NIMAS files but they are not discussed here. For this discussion, we assume you have obtained the prose of the textbook in a word processor format, either RTF data itself, or data for a word processor tool you have available to you (via that tool the prose can be Copied to the Clipboard and then Pasted (as RTF data) into Braille2000). For a large textbook, you might be working with multiple digital files, each covering only a part of the textbook.

The Braille Transcription

In general, each braille volume will involve one or more *transcriber* pages, one or more *preliminary* pages, and a number of *text* pages. There are sometimes volumes that are all glossary or all index, but we are not talking about a book of that magnitude. All braille pages have (braille) page numbers in the lower right corner of the page (interpoint braille may not show them on back side (even numbered) pages) and most braille pages have a print reference number in the upper right corner (transcriber pages do not). The braille page numbers form three sequences, of t-numbers, of p-numbers, and then of (plain) numbers.

There is no concept in the print file relating to braille page numbers: all control for that aspect is by %-code or is done later in Braille2000. How print reference numbers are handled can vary.

As mentioned previously, the page numbers (if any) in the RTF file (in a page header or footer) are not available for processing by Braille2000, and depending on how the RTF file was obtained, the page numbers may not agree with those of the published copy of the book! What you need to observe is whether your RTF file has hard page breaks uniformly in precisely the same place as the page turns in the published book (if so, you are fortunate!). In the typical RTF file, the pagination of the file has no resemblance to the page turns in the book itself.

If your RTF file has hard page breaks (how do you tell? switch Word to Normal view and look... in Normal view hard page breaks are explicitly shown (ignore soft page boundaries)), make sure they match the position of page turns in the print copy. If they agree, then you can tell the Braille2000 import mechanism that hard page breaks signal a change of print page and that the print reference number should be incremented by one. You do this with the %-code %page=npp ("page break means next print page") and you start the numbering using %pn=number ("print number equals the given number") such as %pn=i or %pn=1 (subsequence hard page breaks will increment that number).

If your RTF file does not use hard page breaks to faithfully match page turns in print, then do not use %page= and instead put %pn=number just before the first word of the prose from each page in the print copy (yes, this is tedious, but there is no data to automate this task). If the next print page number is just one higher than the previous print page number, you can insert %np ("next print page") instead of %pn=number. You can also wait and do this in Braille2000 as you proofread the transcription (at that time, you right-click at the first word on each new print page and select "Next Print Page", or you can make yourself a Speedbraille™ Key to do this for the current cursor position).

Braille Page Break

In your transcription, there are certain items that should always begin on a fresh braille page. The %-code %page (not the same as %page= mentioned above) generates a page break in braille and what follows will be on the top of a braille page. For your textbook transcription, you will want new braille pages for (a) the title page, (b) the special symbols page, (c) the transcriber notes page, (d) the contents page, (e) an optional foreword page or acknowledgements page, (f) the first text page, and (g) perhaps at major book boundaries (sections and/or chapters). Simply locate those places in the RTF file and insert %page (typically on a line by itself).

Running Heading

The concept of running head is not unique to braille but in braille the length of the heading is severely limited and so special considerations apply. The RTF file may or may not have a heading, but any heading (in the reserved heading area of each page) is ignored by Braille2000. To assert a braille running head, copy the heading to the Clipboard and then paste it into the document as regular prose and put %runhead above it on a line by itself. Shorten the prose to what will fit in about 30 cells. Do this *after* the prose for the braille title page (discussed below), because the title page does not show a running heading (your running head should start on braille page two).

The First Text Page

After the transcriber pages and preliminary pages you will have your first text page, and by convention it begins with the full title of the book. If the full title is short enough, the full title will have been used for the running head for the transcription, and if so, it will *already* be at the top of the first text page. Or perhaps you are transcribing the book without any running head, and in that case you can put %head on a

line above the full title to make it be a centered heading. But if the full title is long, your running head will be some short sub-phrase and you won't want that at the top of the first text page. In that event, to begin the first text page, use %pager rather than %page. The added letter "r" causes the new braille page to be prepared without a running head (as an exceptional page, the running head will resume on the subsequent braille page).

Also remember that at the first text page, you will need to change braille page numbering to plain numbers (you can use %bn=1) and you are likely to need to state the print page number (you can use %pn=number). These %-codes are discussed further below. But we jumped ahead to explain how the running head works... the textbook transcription generally involves the following segments:

1. The title page
2. The special symbols page
3. The transcriber's notes page
4. The table of contents
5. Other optional preliminary pages
6. The first text page
7. Many additional text pages
8. An optional glossary
9. An optional index

Of course each textbook may have more or less of these things and/or other kinds of material beyond the standard things we discuss here. In the segments below, we explain how to manage RTF data to handle each of these book segments.

The Title Page

The braille title page is discussed at length in *Braille Formats Principles of Print-to-Braille Transcription, 2016*. You may find it handy to create a RTF template file of a standard title page, such as what we show here:

```
%pagef
Title; Edition; Grade Level
Subtitle; Series
%space
Author, Degrees
%space
Published by Name of Publisher
City, State of Publication
publisherwebsite.com
Copyright Year by Holder of Copyright
Further reproduction or distribution in other than a
specialized format is prohibited.
Transcription of ISBN
Printing history: 9 10 071/55 07 06
%space
Transcribed year into Unified English Braille by
Name of Transcriber
Tactile graphics by Name of Tactile Graphics
Specialist
Affiliation of transcriber
City, State, Postal code abbreviation
%space
In 1 Volumes
Volume 1
Braille pages t1-t7, p1-p14, and 1-80
Print pages i-xx and 1-a60
```

The standard title page has five blocks of information separated from each other by one or more blank lines, arranged so that the title page *exactly* fills the first braille page. So, given the print information, how do you know how it will fit a braille page? Well you don't. But you hope that with just one blank line (denoted by %space in the template) it will be less than a braille page, and you use the "fill" option (the "f" on %page) to add just the right number of additional blank lines to make it perfect.

%pagef (like %page) starts a new braille page, and the "f" declares it to be a *filled* page... if it is short, more blank lines are added dynamically (bottom up) at existing blank lines to make it big enough to fill the page. In Braille2000 you can use Page Control to create a filled page.

By default, Braille2000 ignores blank lines during import. %space is used to create a blank line that will persist. You could also use normal blank lines if you added %pbl ("preserve blank lines") at the top. But if you do that, blank lines later will also be

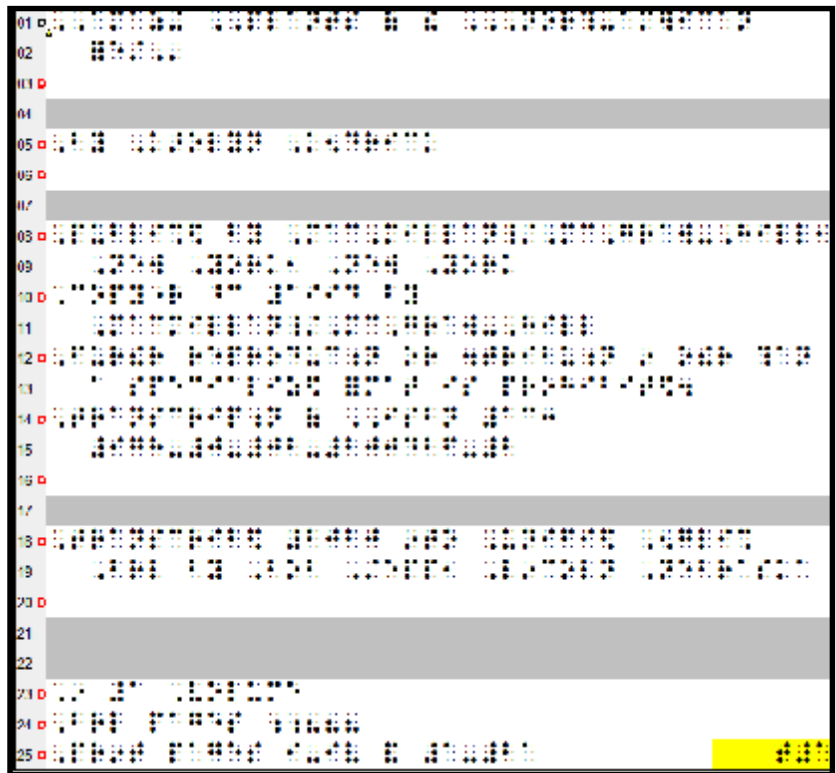
imported, so you might want to add %pbl=off at the end of the title page to stop the effect. Using %space seems easier, but it is a matter of taste.

```

%pagef %bn=t1
AMAZING PLANTS of the NORTH-AMERICAN
FOREST
%space
By Karolyn Kendrick
%space
Published by MacMillan/McGraw-Hill, New York,
New York
Copyright © 1994 by Macmillan/McGraw-Hill
Further reproduction or distribution in other than a
specialized format is prohibited.
Transcription of ISBN 13: 978-0-02-200426-2
%space
Transcribed 2020 into Unified English Braille by Bob
Stepp, Lincoln Nebraska
%space
In 1 Volume
Braille pages ???
Print pages i-iv and 1-21
%runhead
AMAZING PLANTS
%head %page
TRANSCRIBER'S NOTES

```

At the left is a title page prepared for the children's book *AMAZING PLANTS of the NORTH-AMERICAN FOREST* by Karolyn Kendrick. It has the five blocks of information as pertains to the book. The first line uses %pagef to tell Braille2000 to make a filled page and it uses %bn=t1 to start a braille page number sequence for the transcriber pages. Near the bottom of the excerpt, %runhead is used to declare a running heading that is to begin on braille page two. After that, %head causes "Transcriber's Notes" to become a centered heading, and %page places this at the top of a new braille page. When this prose is imported into Braille2000, the following first braille page is the result.



In the result window (right), the gray lines are the from page filling. The white blank lines are from %space. The 1-3 paragraphs came from the title page prose (above) using paragraphs with a hanging indent of 0.25". Notice that the page number is t1.

The next braille page has "Amazing Plants" as the running heading and then "Transcriber's Notes" appears as a centered heading (with a blank line before and after). The special symbols page normally precedes the

transcriber's notes page... with Braille2000 the special symbols page is automated: we will insert it last and this is discussed later in this paper. The next page in our sample textbook is contents.

The Contents Page

The table of contents for our sample book looks like this. Like most tables of contents, each entry consists of a title followed by leader dots and ending with a page number. The leader dots are not regular prose, they are a formatting option for a tab skip: the page numbers are offset from the titles by one or more tabs. Braille2000 finds the page number for each entry based on there being a tab just before the page number. The transcriber added the caption "Chapter". Braille2000 has special formatting automation for contents, as signaled by %toc before the contents material and %toc=off afterwards. The first line has %-codes to set the print page number (%pn=i), to set the start of preliminary pages (%bn=p1), to make "Table of Contents" into a centered heading (%head), and to start a new braille page (%page).

```

%pn=i %bn=p1 %head %page
Table of Contents
Chapter
%toc
Introduction..... iii
Chapter 1 All-Important Plants..... 1
Chapter 2 A Plant's Many Parts .....5
Chapter 3 Plant Adaptations ..... 9
Chapter 4 Feeding the World..... 15
Conclusion.....18
Glossary ..... 19
Index .....21
%toc=off

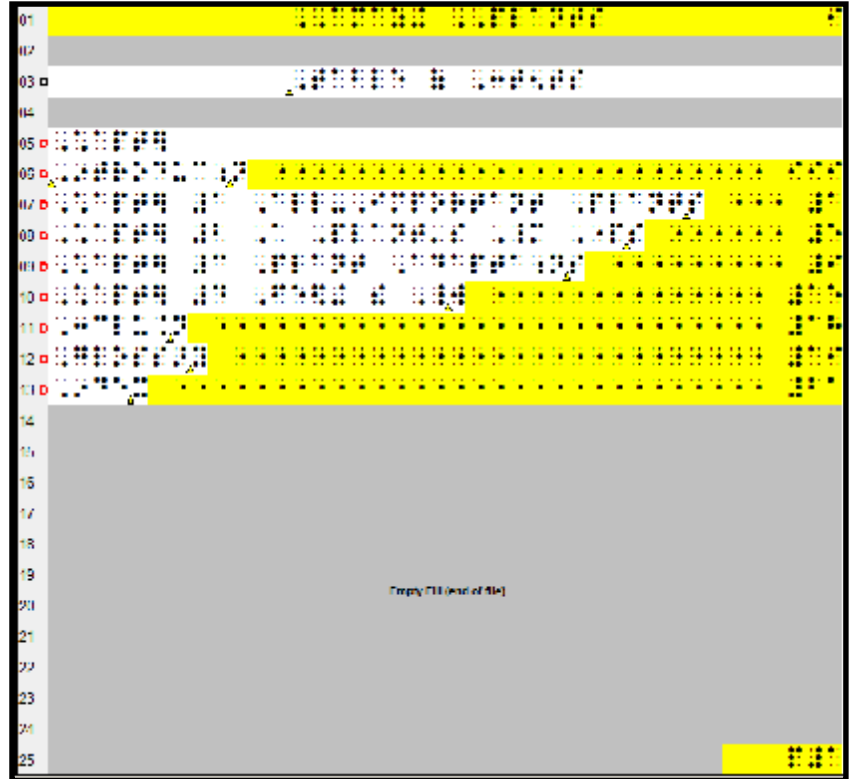
```

In Braille2000, contents page format is handled by the Line Number tool.

The resulting braille page shows here:
This page has print number i and
braille number p1.

The various small yellow triangles denote *annotations* (markup controls) with settings for the page layout. In line 03 the triangle represents three annotations: for page break, for the print number, and for the braille number.

On lines 06 to 13, the triangles near the end of the prose are declarations of the line's line number. On line 06 the triangle at the beginning of the line is for layout control (this page has a right margin of six cells). Another layout control annotation (setting the right margin back to zero) will occur at the top of the following page.



The yellow dot-5 guide dots are dynamic: you can edit the prose in the table of contents and the number of guide dots will adjust accordingly. If you wish to change an element showing in yellow, such as the running head or a line number, point the mouse at the element, click the right-hand mouse button and select "Change from inception".

Other Optional Preliminary Pages

`%page %pn=ii`

`%subhead`

Introduction

Think of all the things you need to survive. Are food and oxygen near the top of your list? Well, thank a plant for both. Plants work from sunup to sundown to provide the food we eat and the oxygen we breathe. These silent, green workers have the most important job in the world. They use the sun's *energy* to make food. Sooner or later, food from plants ends up feeding us. Plants also give us clothes, houses, medicines, fuel, and much more. Even this book is made from plants.

We can use plants in many ways because there are so many types of plants-more than 288,000 and counting. The plant *kingdom* includes giant trees, like the 300-foot tall (91 meters) sequoia, and tiny plants like the watermeal, which is smaller than a pencil eraser. Plants live on every continent, even icy Antarctica. They thrive in dry deserts and wet rainforests. Some plants even live in saltwater.

In the sample book there is an introduction on page ii, i.e., it is preliminary page. You can manage a page like this by using `%page` to start it on a new braille page with `%pn=ii` to give its print number. Use `%subhead` to set "Introduction" as a cell-5 subheading.

Instead of using `%subhead`, you could adjust the paragraph properties of "Introduction" to give it a left margin of 0.5" and spacing before of at least 12 pts (with those settings implicit formatting would generate a cell-5 subheading).

The braille for "Introduction" will be bold.

The First Text Page

The first text page also begins on a new braille page (like the Introduction above), but there are additional elements to manage. The first text page will be the first of plain braille page numbers (use `%bn=1`) and its

centered heading will be the full title of the book and should be at the top of the page in lieu of the running head. Use %pager instead of %page to suppress the running head on just this one braille page.

```
%pager %pn=1 %bn=1
%head
AMAZING PLANTS of the NORTH-AMERICAN FOREST
%head
Chapter 1
    Important Plants All living things need energy. We
    all get energy in different ways. When you need an energy
    boost, you probably grab a snack. You know that food
    gives you the energy to get back to work or play. Food is
    our energy source. Animals also eat plants or other animals
    to get the energy they need.
    Plants are different. They make their own food, using
    sunlight for energy. They transform light energy from the
    sun into food energy. So when you bite into a juicy orange,
    you are actually tapping into the sun's energy. Energy from
    the sun is in all of the foods we eat, thanks to plants.
...

```

The RTF data for the first text page is adjusted as shown here. The full title is pasted in preceded by %head (it will be a centered heading. %pager causes this material to start on a new braille page without a running head. The print page number will be 1 and the page will be the first of regular braille page numbers (via %bn=1).

"Chapter 1" will be a centered heading. The paragraphs (via implicit formatting) will be 3-1 indented braille paragraphs. Otherwise you could use %indent1 to format them.

In Braille2000, the effect of %pager is handled by Page Control.

The Glossary

```
%head %page
Glossary
%footer=1
adaptation a change of a physical feature or behavior to fit
    environmental conditions (page 12)
cell the tiny building blocks that make up all living things
    (page 4)
%gw2 cell membrane a thin membrane that surrounds
    every living cell, separating the cell from the
    environment around it (page 5)
chlorophyll a green substance in plants and other
    producers that traps light energy that plant use to
    make food through photosynthesis (page 6)
chloroplast a part of a plant that changes light energy into
    food energy (page 5)
ecosystem all the living and nonliving things in a certain
    place and their interactions (page 12)
...
%footer=off

```

A glossary is a list of word definitions, such as shown to the left. Glossary pages show a guide word footer on the last line of each page giving the first and last keywords on the page. This is automated in Braille2000.

This page begins like others, using %page and %head so that "Glossary" is a centered heading at the top of a braille page (following any running head). The %-code %footer=1 establishes an automated guide word footer that picks up keywords from cell 1 on the braille page. The print paragraphs have a hanging indent of 0.25" and by implied format will be 1-3 paragraphs. After the glossary, %footer=off stops showing the guide word footer.

The automated footer picks up individual keywords at the given indent position (cell 1 or cell 5 as specified in %footer=n). In the entry where the term is "cell membrane", the %-code %gw2 indicates that the next two words *together* constitute the keyword. If a glossary's keywords are not consistently in cell 1 or cell 5, you can use %footer=0 to show the guide word footer without any automatic pick-up of any keywords. In that case, for each entry, you must place %gw in front of each keyword. (%gw is the same as %gw1; use %gwn in front of a keyword phrase composed of *n* words.)

An Index

If the book has an index, it can usually be done as a list. Use %list to format a one-level list as 1-3 braille paragraphs. For an outline or multi-level list, use %list1 for the first level items, use %list2 for the second level, etc. up to %list9 for the ninth level. Braille2000 sets the runover in a multi-level list uniformly to the maximum indent plus two cells.

The Special Symbols Page

So far we have neglected to do a special symbols page. It normally precedes the transcriber's notes page. We do it last because the special symbols page is automated in Braille2000, and only when the transcription is essentially done do we know which special symbols were actually used in the manuscript.

To automatically generate the special symbols page, put the cursor where the page should be inserted: normally you should put the cursor at cell 1 on the first line (the heading) of the transcriber's notes page. Then click the Import button and select Insert and Template File. In the "open" dialog box, you should see "UEB Special Symbols" (with or without the .rtf suffix). Open this file. The page (or pages) of special symbols list will be inserted!

Special-Purpose %-Codes

Emphasis Transfer

Not all kinds of print emphasis are important to the transcription. This is a subjective determination by the transcriber. The %-code %emp=*p*=*b*, where *p* is replaced by one or more of the letters ibup and *b* is coded similarly, manages emphasis during import. *p* denotes a kind of print emphasis (i is italics, b is bold, ib is bold italics, p is plain, etc.) and *b* denotes the imported braille emphasis. For example if all bold in print should be imported as italics, you can use %emb=i. This will convert bold, bold-italics, bold-underlined, bold-underlined-italics in print to italics in braille. If you use %embi=u then bold-italics and bold-italics-underlined would become underlined in braille (but bold-non-italics would remain bold). You can use %emb=p to cause all bold in print to become plain text in braille. In some cases, you may need multiple %emp=*p*=*b* codes to cover all the possibilities.

Hidden Prose

If there is a portion of the document that you do not want imported, use %hide at the start of such material and %hide=off at the end. When "hidden", the prose does not import.

Transcribers Note

Put %tnote on a line by itself preceding a paragraph that is a transcriber's note (tn). The system will insert begin-tn and end-tn indicators and arrange the prose in a 7-5 braille paragraph.

Inline Transcribers Note

Put %tnl (followed by a space) directly before an inline tn. Put %tnr (preceded by a space) directly after the inline tn. These %-codes generate the appropriate tn indicators.

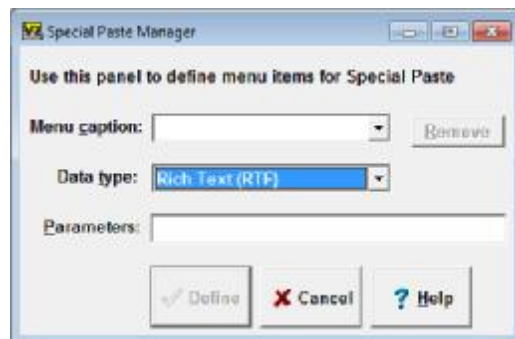
Code Selection

You can insert %L (written here using capital L to clearly show "L" rather than "l" (lower case) that looks like "one"... lower case l works just the same) to select the braille code (previously called "language" in Braille2000 Version 1) to be used during import. You follow %L by code-selection keywords separated by commas (%-codes cannot contain spaces). For example, %Lenglish,uncontracted (or more specifically %Liceb,english,uncontracted) will select uncontracted UEB. %Lbana,english,contracted will select contracted EBAE. If you are doing a foreign language textbook using World Braille Usage single-cell accented vowels, you might use %Lspanish for the Spanish portions. When you change the code (e.g., %Lspanish) you can change back by using just %L.

If you find you need many code changes, you can define code *numbers*. For example, %L3=spanish will change the braille code to Spanish and it will define code 3 to be Spanish. After that, %L3 is the same as writing %Lspanish. (Code numbers need not be used in numeric order.)

Special Paste

Customized paste behaviors are available via right-click to the Paste button and selecting "Define RTF Paste". This shows the dialog box presented here. Type a menu caption and then enter parameters in the appropriate boxes. Your menu caption will be a menu item when you right-click the Paste button. When you select your menu item, the parameters line you used will be added (on the fly) to the Clipboard data as the first line. You put into the Parameters box one or more %-codes (just as you might put them into the RTF file) that you want added at the start of the data.



For example, if you put %Lspanish in the parameter line, your custom menu item will import Spanish prose using the Spanish braille code.

For example, if you want a way to past in prose without any emphasis, you could put %emb=p %emi=p %emu=p into the parameters box... with that line inserted in front of the Clipboard data, all kinds of emphasis would import as plain text.

Appendix A: %-code Glossary

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
%block ν	To set the following paragraphs as block paragraphs of level ν
%bls	To turn on blank line separators, added whenever the indent of a new paragraph matches the runover of the previous paragraph. Use %tb to turn this off.
%bn= <i>number</i>	To give the braille page number in textbook format (lower right) The number may have a letter prefix of t or p to denote transcriber or preliminary pages, respectively. If you use this directive do not use %ln= described below.
%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=1 %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.
%footer	To cause the single following paragraph to be used as the running footer
%footer=off	To stop showing a running footer
%footer= ν	To activate an automated guide-word footer, ν can be 0 or 1 or 5 If ν is 1 or 5, the guide words are the first word in cell-1 or cell-5 paragraphs on the page, respectively. If ν is 0, all guide words are indicated via %gw (see below).
%g ν	To change the language (braille grade) to number ν , where ν 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) 3: computer braille code (CBC) only in BANA context %g without a digit returns to the previous code (same as %L) Note: for a document in a foreign language (always uncontracted), 1 and 2 do nothing
%gw ν	To indicate what word (or words) are to be guide words Place %gw (set off by a space before and after) immediately in front of the word in a glossary entry that is to be used as the guide word. If more than one word is to be used, give the number of following words, as in %gw3. If you have used %footer=0 then you must use %gw to tag all guide words that you want to be active. If you have used %footer=1 or %footer=5 then you can optionally use %gw to indicate alternative and/or additional guide words that are active.
%head	To cause the single following paragraph to be centered with a blank line before and after

<code>%headv</code>	To cause the single following paragraph to be a heading Possible values for v are 0,5,7 and they generated centered, cell-5, and cell-7 headings, respectively. (When used as <code>%head5</code> or <code>%head7</code> the operation is identical to <code>%subhead5</code> and <code>%subhead7</code> , respectively.)
<code>%hide</code>	To cause all following text to be ignored (hidden from the transcription)
<code>%hide=off</code>	To terminate <code>%hide</code> , subsequent text will not be ignored
<code>%indent</code>	To set the following paragraphs as 3-1 braille paragraphs
<code>%indentv</code>	To set the following paragraphs as indented paragraphs of level v
<code>%interpoint</code>	To set the embossing mode to interpoint (double-sided) When used, this directive should be at the beginning of the document.
<code>%L</code>	To set the language back to its previous value (if any)
<code>%Lkeywords</code>	To set the language via keywords The same behavior as two items down, but without defining any language number.
<code>%Lv</code>	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 (<code>%L0</code> and <code>%L</code> (no keywords) are equivalent and will reinstate the previous language).
<code>%Lv=keywords</code>	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.
<code>%left</code>	To cause all following paragraphs to be regular (uncentered)
<code>%list</code>	To set the following paragraphs as 1-3 braille paragraphs
<code>%listv</code>	To set the following paragraphs as list items of level v, i.e., <code>%list1</code> for the first level, <code>%list2</code> for the second level. This forms a multi-level list (as per 2011 formats) until you use a non-list directive (such as <code>%indent</code>).
<code>%ln=number</code>	To give the braille page number in literary format (upper right) The number may have a letter prefix of t or p to denote transcriber or preliminary pages, respectively. Note that <code>%bn=</code> and <code>%pn=</code> described elsewhere in this section are for textbook format—do not mix the two: if you use <code>%ln=</code> in your document do not use <code>%bn=</code> and <code>%pn=</code> and vice-a-versa.
<code>%note</code>	To set the following paragraphs as 7-5 braille paragraphs
<code>%np</code>	To assert "next print page" in textbook format For this to act you need to have used <code>%pn=</code> previously to assert print page numbers. <code>%np</code> is then used before the first word of a new print page to increment the print page number (and generate a print page turn line as necessary).
<code>%off</code>	To turn off all directive processing in the remainder of the file This directive should be placed on a line by itself. After this setting, <code>%</code> is translated as an ordinary character. And thus there is no way to reestablish directive processing (because all subsequent directives are treated as merely plain text). For the occasional leading percent sign (percent following a space as in <code>%data</code>) add a leading <code>%</code> to handle it as a normal character, as in <code>%%data</code> (see <code>%%text</code> below).
<code>%page</code>	To generate a braille page break Special effects for the new braille page are given by added letters and numbers. <code>%page=</code> is used to define the effect of a hard page break in the print document. (see below)
<code>%pagev</code>	To generate a braille page break to a front or back page position (works only in an interpoint document (see <code>%interpoint</code>) The digit v can be 1 or 2 or 3. <code>%page3</code> is the same as <code>%page</code> . <code>%page1</code> breaks to the next front page (inserting a blank page as necessary). <code>%page2</code> breaks to the next back page (inserting a blank page as necessary).

`%pageh` To generate a braille page break and apply certain optional attributes to the new braille page. Optional attributes can hide certain page elements on the new page and the page can be set as a "filled page" (see below). The letter r hides the running head. The letter p hides the print number. The letter b hides the braille number. (Number hiding does not change the page number sequence.) The letter s suppresses the print number (the new page does not participate in the number sequence, as if it doesn't exist). The letter f turns on page filling (the final prose will always fall on the last line of the page (blank lines are added as necessary to make that happen). When necessary you can follow `%page` with more than one attribute letter and you can also add a digit (1-3) as mentioned above. For example `%page1rp` would break to the next front page position and hide both the running head and the print number.

`%page=npp` To have each hard page break in print imply `%np`. Like `%np` (described above), print page numbering needs to be established using `%pn=`. The document may use both hard page breaks and `%np` directives to advance the print page number, as appropriate.

`%page=bpb` To have each hard page break in print imply `%page`. You should normally use `%page=npp` or `%page=bpb` but not both, however one does not cancel the other: if you use both then each hard page break implies both `%page` and `%np` until you use `%page=ignore`.

`%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 for textbook format (upper right). Note: if also using `%bn=p1` (e.g., to start preliminary pages), put that directive before `%pn=`. To have print page numbers in your document you need at least a starting `%pn=` directive. After that, you need to signal each print page start, either via hard page breaks and `%page=npp`, or by inserting either `%pn=` or `%np` just before the first word of each print page. For unusual print page numbers you will need to use `%pn=` in a manner mentioned just below.

Special Print Numbers

The print number may be in the form prefix-number, as in 10-17 or Intro-1. In UEB (strange as it is) a numerical prefix is transcribed as a preceding number with its own number sign (and no distinguishing punctuation, i.e., it will read 1017 in print or #AJ#AG in braille, unfortunate). But at other times, 10-17 might denote spanned pages and need the hyphen in braille.

When a numerical section number is used (no hyphen in braille), replace the hyphen with the vertical bar (|) as in `%pn=10|17`, this will generate the proper braille number #AJ#AG (no hyphen).

When a hyphen is desired (spanned pages) use the hyphen as in `%pn=10-17`. Additional page number behaviors may be indicated by following the page number with a colon and a special symbol, as follows:

Colon 1: for section-number hyphen single-page-number: `%pn=Ref-8:1`

Colon 2: for spanned pages transcribed as a whole: `%pn=3-4:2`

Colon Letter: to give the continuation letter (rarely needed): `%pn=6:c`

If the page number is non-numerical or mixed, enter it as given, such as `%pn=One` or `%pn=Intro1`. (In some cases the "Next Print Page" function will be unable to increment such a number.)

See Also Print Number Extras (elsewhere in this document).

`%pn=off` To cancel the generation of print page numbers

`%reqsp` To check for a blank line in braille at this point. Same as `%reqsp1`.

`%reqspv` To check for a blank line in braille at this point, generating one if necessary. `%reqsp` is like `%space` except where the blank line would be the first or last on a page or where a blank line already exists by other means.

`%runhead` To cause the single following paragraph to be taken as the running head. Note: be mindful that a running head must be short in order to fit on but one line of braille.

<code>%runhead=off</code>	To cancel the use of a running head
<code>%space</code>	To generate a blank line in braille
<code>%spacev</code>	To generate v number of blank lines in braille
<code>%subhead</code>	To cause the single following paragraph to be a 5-5 paragraph with preceding blank line
<code>%subheadv</code>	To cause the single following paragraph to be a v-v paragraph with preceding blank line
	Use <code>%subhead5</code> to generate a cell-5 subheading. Use <code>%subhead7</code> to generate a cell-7 subheading. (Other values for v also work but are not standard heading choices.)
<code>%symbolslist</code>	To declare the following items to be Special Symbols List template data
	See discussion at Automatic Special Symbols List (elsewhere in this document).
<code>%symbolslist=off</code>	To terminate the section of symbols list template items
<code>%tb</code>	To start “tight blocked paragraphs” i.e., to turn off <code>%bls</code> (blank line separators between certain paragraphs). Beginning in version 2.269, this option is on by default.
<code>%tnl</code>	To generate a beginning transcriber's note indicator in braille
<code>%tnote</code>	To set the following paragraph as 7-5 braille transcriber note with beginning and ending indicators
<code>%tnr</code>	To generate an ending transcriber's note indicator in braille
<code>%toc</code>	To set the following paragraphs as contents entries
<code>%tocv</code>	To set the following paragraphs as contents entries of level v (v=0,1,2,3, etc.)
<code>%toc=off</code>	To end special handling for contents entries
<code>%x-y</code>	To set the indent (x) and runover (y) properties of the following paragraphs
	For example, to achieve a standard 3-1 paragraph style, use <code>%3-1</code> .
<code>%%text</code>	To use the % character as plain text, use <code>%%</code> to represent %