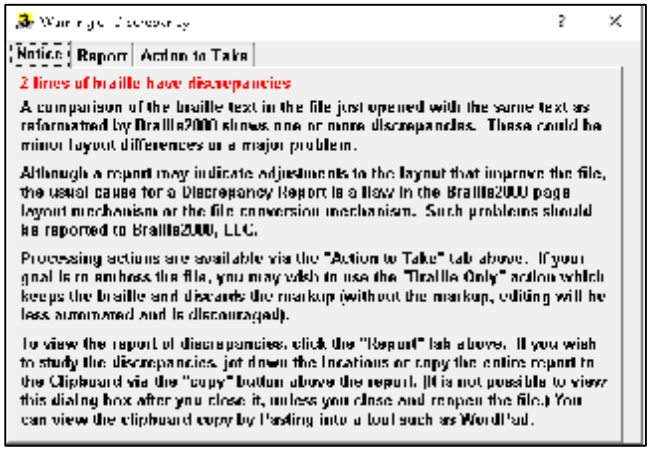
Braille2000, LLC
6801 Southfork Cir, Lincoln, NE 68516
Phone: (402) 423-4782
http://www.braille2000.com

Everything You Always Wanted to Know about

Discrepancies*

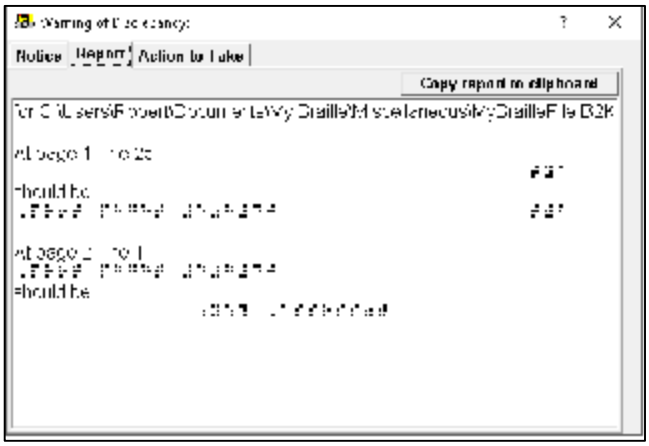
*but were afraid to ask

With deference to David Reuben, there are certain events that are at least mildly traumatic, and one of them is the dreaded “Discrepancy Report” in Braille2000. Fear not. It’s not the end of the world, as is explained below.



From time to time, perhaps when you least expect it, you may see this box when you open a braille file. As shown here, the “issue” has to do with 2 lines, but in some cases, the number may be alarmingly high, perhaps hundreds of lines! Do not panic. (Note that older versions of this box mention the old business name “Computer Application Specialties Company”, ignore that.)

Via the “Report” tab, you can see what it is that is “wrong” with the file. The report could be many pages long… do not worry about its length. You can browse the report to know what the issue is about.



In this case the report has only 2 “should be” statements. There could be hundreds. But why are there any? It all has to do with the Braille2000 .b2k type of file, a unique kind of file with useful properties.

What you might not know is that the braille “engine” inside Braille2000 handles only one type of file, and it is *not* .b2k. Who knew? Braille2000 operates only on .bml files, and yes, you could save your work using .bml type files, but most users don’t. And .bml files cannot have discrepancies. But for pragmatic reasons of utility, you probably should continue using .b2k files. If you do get a discrepancy report, the usual response is to “Accept” the file and continue your work

To understand the notion of “discrepancy” you need to know a bit about the file types. The .bml file (what the engine works with) is a text file using XML internal syntax as “markup” describing phrases in ASCII-braille notation. It contains braille cells, but also lots of non-braille stuff to manage the layout and other automation. Many text processing tools use XML notation (e.g., Word and NIMAS publisher files and BrailleBlaster files). XML is an appropriate notation for document automation. (Older text processing tools such as WordPerfect and Duxbury DBT and Megadots and Wordpad, successfully use non-XML markup. For example, “Rich Text” (.rtf data) is quite unlike XML, but is still used by modern Word and Windows systems).

In the .bml file, braille prose is intermixed with markup. This is true for all documents using markup. The file only makes “sense” when handled by the right tool… by Braille2000 in the case of .bml data. Without Braille2000, it’s opaque, essentially useless. The .b2k file is not like that. It is unique among markup files.

The .b2k file consists of two parts. The first part is page-layout ASCII-braille, i.e., lines of text in typically a 40x25 layout, just like a .brf file, that if you send it to a traditional embosser (loaded with the correct size paper), you will get nice braille. Furthermore, that data can be delivered to a refreshable braille notetaker device. The first part of a .b2k file is just braille. No markup. Then there is a second part that is just XML, the same XML as in the equivalent .bml file. The boundary between parts one and two is indicated by one Ctrl+z character, because this marker character is honored by the old DOS command called COPY. Because all Windows versions offer the COPY command, anyone holding a .b2k file can extract the braille portion using “COPY /A file.b2k file.brf” where /A tells COPY to honor the Ctrl+z mark for “end of data”, and you mention two file paths, the first being the input and the second being the output. It does not matter to COPY what the file names are: all that happens is that the data of the first file is copied to the second file, halting at Ctrl+z. And by the design of the .b2k file, this operation extracts the first part, the page-braille portion, data like a .brf type of file. It is for this kind of open access that .b2k files have high utility and popularity: the .b2k file is unique in how usable braille can be easily extracted without any special software.

Because the second part of a .b2k file holds all the markup of the equivalent .bml file, it is a quick and simple process to convert .b2k data to .bml data, and this is what happens every time you open your .b2k file: it is converted, on-the-fly, to .bml form and that is then opened (remember: the “engine” inside Braille2000 only handles .bml files; all other types undergo on-the-fly conversion to .bml form). But how does this neat trick relate to discrepancies? It’s because the .b2k file holds the braille in **each part!** The two parts hold it via different notations. Normally the two parts yield exactly the same braille, but if they don’t… bam! You have a “discrepancy”. Thus, the occasional discrepancy is the price you pay for using the .b2k file with universal accessibility.

As far as we know, no other word processing tool other than Braille2000 has anything like discrepancies. Once you have a file (i.e., .b2k) with the same prose expressed twice, you have to account for the situation in which those two versions differ. Of course, one would expect both parts to give the same braille, and normally they do. Why would there ever be any difference? It has to do with static page layout versus dynamic page layout.

Braille2000 is a unique WYSIWYG (“what you see is what you get”) braille text editor, like Word in this regard (Duxbury DBT and BrailleBlaster are not WYSIWYG). A WYSIWYG tool manages

document pages that dynamically change layout as you edit the document. The markup elements (in the second part of the .b2k file) provide the “controls” (e.g., styles) for various aspects of page layout. Braille2000 has general algorithms for braille page arranging that are adjusted and tuned by document settings. In the normal case, braille pages have been arranged appropriately when the file is saved. When saving a .b2k file, the page-braille (braille only) part is written, and then Ctrl+z is written, and then the XML markup is written. At that moment, the markup (that dynamically arranges pages) and the static page-braille are exactly the same: the former *generated* the latter. All is well until something changes, typically the algorithmic part of the Braille2000 page layout mechanism.

As long as Braille2000 remains unchanged (and barring bugs), a .b2k file opens just fine without any discrepancy. Although not essential, whenever a .b2k file is opened, Braille2000 loads the page braille, and converts .b2k to .bml form, and loads the .bml prose using XML markup. The loaded .bml data dynamically regenerates all braille pages (and page layout updates continuously as you edit the document). Braille2000 immediately compares the just-regenerated pages to the static page-braille from the first part of the file. They ought to be identical. If they are not identical, each braille line that is different is a “discrepancy”. And you may ask, why report it at all?

No word processing tool is perfect from first release. All tools (Word, Duxbury DBT, BrailleBlaster, etc.) occasionally receive updates that make subtle changes to document layout. But only Braille2000 (when using .b2k files) has any means to detect any changes: in all other cases layout changes are unannounced and unknown. That is also the case when using Braille2000 .bml file types. There is one situation in which it is important to know if the layout has changed: when a braille printing house goes to emboss replacement pages for an existing braille volume (e.g., replacing a torn page, a coffee spill, etc.) you want replacement pages that are dot-exact duplicates of what was done previously. A braille-only file (such as .brf) would be unchanging. But any file with markup (Braille2000, Duxbury, Megadots, BrailleBlaster, etc.) would not necessarily offer dot-identical copies over time (when the software has changed due to software maintenance). So, for this need (braille printing houses), Braille2000 announces any discrepancies it discovers (they can occur only in .b2k files). And the user can then choose to work with either the static page-braille (the “Braille Only” action) or the dynamic braille provided by markup (the “Accept” action). For most uses, the Accept option is the right choice. When you Save the file, the static and dynamic parts will be re-synchronized and discrepancies will no longer occur.

Discrepancies normally occur only when the Braille2000 page layout algorithm has been corrected (i.e., a layout bug is fixed) and a newer (i.e., better) version of Braille2000 is opening a .b2k file saved by an older (i.e., less-good) version. Unless you need a dot-identical copy, you want to Accept the new and continue, because the new (dynamic) layout is normally the better.

Some differences in page layout can affect the amount of prose that can fit the page. If that changes, lines can get “pulled up” or “pushed down” throughout the file, and that can lead to hundreds of secondary discrepancies because page lines (static vs. dynamic) will not match. Thus, the number of reported discrepancies is not of itself an alarming factor.

Recommendations

If you find discrepancies upsetting, you can do your work using .bml Braille2000 files (and as needed, save a .brf static version). .bml files cannot have discrepancies, but you can’t get plain-braille (as for a refreshable notetaker) from them unless you have Braille2000.

Most users will want to use .b2k files. If you happen to get a discrepancy report, the normal response is to Accept and then Save the file. It helps if you and your colleagues use the same version of Braille2000. (Layout algorithm bug-fixes are rare, but versions that differ by many years may have differences that manifest as discrepancies.)