THE BRAILLISTS FOUNDATION

WHAT IS A BRF AND WHY WOULD YOU WANT ONE?

Matthew Horspool: Welcome back to Braillecast Extra. Coming up this time, What Is A BRF File And Why Would You Want One, a session recorded on Tuesday, 4th May 2021, presented by me and introduced by Dave Williams.

Dave Williams: Good evening and a very warm welcome back to the Braillists Foundation and our regular Tuesday evening Master Class. This evening Matthew Horspool will be introducing us to the subject of BRF. Why would you want a BRF? How would you create a BRF? What might you do with a BRF? Matthew will be guiding us through all of that shortly and taking your questions of course as well.

As always, on moderation duties this evening, is Ben. Good evening to you, sir.

Ben Mustill-Rose: Hello. Going to be an interesting one this evening, I think, actually.

Dave Williams: Yes, indeed, bit technical possibly. Matthew will let us know shortly but I think lots of practical application as well.

Ben Mustill-Rose: Yes. We were talking about this earlier actually and I was saying, it's one of those things where you maybe don't think you need it but when you actually discover what BRF files are used for and how versatile they are, I would say that then, they do actually come in quite useful, but I guess that's a bit of a spoiler, all will be revealed shortly.

Dave Williams: So, we'll get to how to raise hands and all that good stuff when we get round to questions. There is quite a lot of content to get through tonight.

Before we jump into that, a reminder of some of the other things that are happening around the Braillists. Don't forget, of course, you can keep up to date with all our news through the Braillists newsletter, so any last minute changes or if you'd like a heads-up on what topics we're going to be covering in future weeks, for example in two weeks time, we're hoping to be bringing to you a session about Braille on Android, a subject we get asked about quite a lot. So, if you'd like to know what's coming up, do sign up for our newsletter, if you're not already. You can do that from the braillists.org website.

A reminder about our Thursday night Braille book club. That starts at the slightly earlier time of 6pm. We have rooms for different reading abilities, beginning, intermediate and more experienced Braille readers. You're very welcome to join the book club.

Next week on Tuesday at 7:30, we'll be reopening our Braille Bar which is an open opportunity for you to ask any Braille-related questions, whether that's about code, kit, teaching or something that we haven't thought of. So, anything Braille-related and we usually have a little panel that we put together who will be very happy to take your Braille-related questions and that's next Tuesday at 7:30.

Don't forget, of course, these sessions are run in line with the Braillists Foundation moderation policy, so we do try to keep one person at a time speaking so everybody's not talking over each other which makes it a lot easier to hear, especially for people dialling in on the telephone.

When we get to questions, you will be invited to raise your hand and then Ben will acknowledge the hands and you'll be able to put your question to Matthew.

So, without any further ado, I'm going to introduce our good friend and treasurer of the Braillists Foundation, somebody with a considerable experience in the subject of Braille transcription, also chairs the Braille General Group at UKAAF and commentated on last year's International Council on English Braille's General Assembly when that was hosted virtually from the UK. Mr. Matthew Horspool, good evening to you.

Matthew Horspool: Good evening, Dave. You make me sound terribly professional. I'm not really used to it. I'm still used to thinking I'm just this person that does a bit of Braille here and there.

It's wonderful to be back with everybody. I've presented a couple of Master Classes for the Braillists and a lot of you will know me from other Braillists calls and the Braillecast podcast and the help address and one or two other things like that.

In this session, we're going to talk about BRF files. Hopefully that's what you're expecting. If you're not expecting BRF files, then hang around anyway because you might find the session interesting. If you are expecting BRF files, hopefully you'll also find the session interesting.

Just to give you a quick summary, there's going to be probably quite a lot of talking today. I'm really sorry about that but it's one of those things where you think, "Oh, it's just a BRF file. What could you possibly say?" and then you write down a list of what you're going to say and the list just gets longer and longer and longer.

I'm going to try to keep it as snappy as I can. If there's time to run a couple of demos, I'll run a couple of demos but I am just going to try and get through the information and give time for questions and maybe there'll be the potential to run some of the demos. If we don't get to them on the session tonight, I'll run some of those demos and publish them on the Braillecast podcast in due course.

We're going to talk about what a BRF file is. We're going to talk about: why we need them and some typical uses of BRF files; characteristics of BRF files which sounds awfully technical but it's essentially what are they, how are they made up underneath the hood; some of the limitations of BRF files; how to obtain BRF files and how to make your own; how to create them from scratch; reading BRF files; navigating BRF files; embossing BRF files and finally a short section about Duxbury and BRF files because Duxbury has some very interesting BRF logic.

I'm just going to talk for about 25 minutes, half an hour, and see how far I get and then we'll evaluate how much further we can go and maybe take some questions.

What is a BRF file? Well, I notice we've got James Bowden on the call and he might wish to come in a bit later on to correct me on this. I don't actually know where BRF came from. We now know it as Braille Ready File and there is a handout for this session and I've put Braille Ready File in the handout because it's the best definition that I've got. Sometimes people call it Formatted Braille but whether Braille Ready File came first and then they shortened it to BRF or BRF came first and then somebody shortened it to Braille Ready File, I don't know.

It's a text-based format. It's been around for a lot of years, probably since about the 1980s or 1990s, at least, and what it allows us to do is to store and retrieve combinations of Braille dots in the same way as a plain text file allows us to store and retrieve combinations of print letters and numbers.

So, what we're essentially saying is it's Braille first. This is not a file like a Microsoft Word file that has everything in print and then you have to translate it. This file has Braille first.

There's no official standard for BRF. Everybody thinks it conforms to the BRF standard. Actually there's isn't one. You can make any sort of file and call it a BRF file if you want. However, it's being used by transcription companies literally all over the world, RNIB use them, APH, National Braille Press, Vision Australia, RNZFB out in New Zealand, everybody uses them and they have been exchanged all around the world and the Marrakesh Treaty will make that a lot easier. So, over time some conventions have emerged and most people do conform to those conventions. So, although there's no standard as such, there is a de facto standard and it's a good idea to follow it because otherwise Braille hardware might not play nicely with them.

There are other formats that are similar to BRF that we won't cover but I do just want to bring them up very briefly so you know what they are. There's a format called BRL. BRL and BRF, I have to say, I don't know what the difference is. I think this is where having no standard really kind of comes into its own.

When I first started transcribing Braille, I used a program called Cipher and Cipher saved what it called BRL files and when I renamed those BRL files to BRF, they worked exactly the same as a BRF file would work. So, as far as I'm concerned, there's no difference. I think there are some differences some of the time in that BRL doesn't contain as much formatting information sometimes as BRF does but really it's just a file extension that somebody decided to use one day because they didn't like BRF, as far as I can tell.

Then we have the portable embosser format, or PEF. This is a standard that I really wish had taken off. The thing about PEF is that it contains a lot more information in a much more structured way than a BRF file does. So, it contains metadata about, for example, whether the book is in multiple volumes, which volume you're in, what size page is this book for, all this sort of thing. It also uses Unicode Braille patterns instead of ASCII Braille patterns and we'll get onto more of those technicalities a bit later on. Unfortunately what that means is that PEF files are very large and quite unruly and they seem to have fallen into disuse. I'm not sure that the standard was ever widely adopted but it really has become a bit redundant now.

So, I've said that a BRF file is Braille first. Why do we need it? We don't have a French first file format. We just write in Word. We don't have an Arabic first file format. We just write in Word. Why do we need to worry about it?

The reason is because in Braille, there is not a straightforward one to one mapping of characters. So, what I mean by that is, for example, in print a number one is always a single character and it's always the number one, it looks like a number one. In Braille, you would argue that the number one is two characters. It's the number sign, followed by the letter A. However, it's not strictly true that the number one is always a number sign followed by the letter A. What happens, for example, if we're Brailling the number 11. We don't Braille number sign, A, number sign, A. We just Braille number sign, A, A.

The same thing happens in reverse. The word "the" in print is three characters. It's T, H and E. In Braille it's only one character. It's the THE contraction, dots 2 3 4 6. But it's not as straightforward as saying, well, this is the THE contraction, whenever you see THE, you must render these four dots, because what about words like "northeast", where you have to use the TH contraction followed by the EA contraction to show that the first half of the word is "north" and the second half of the word is "east".

There are Braille fonts for computers. They do exist and they do work well. But fonts are not intelligent. Fonts just display what they're told to display and there is no way in which you can codify all of the Braille rules into a font. This is not the same as in French. If you're writing in French, for example, and you have an e acute, it's always an e acute, there's no content in which you'd go, "You know what, an e acute isn't written like this." The e acute is always written like an e acute, so a font can do its job.

What we also need in addition to talking about translating print into Braille is a foolproof way of writing Braille itself. So, for example, if you're a Braille teacher and your student wants to write Braille electronically, on a Braille note taker or a Braille display or on a computer, the Braille teacher needs a foolproof way of looking at the contractions that the student has used, so that the teacher can then say, "Okay, you didn't use the THE contraction here but you could have done," or, "You didn't use the WH contraction here but you could have done." We need a way of being able to codify that information.

Similarly if you're taking notes, you might miss out a contraction or add an extra contraction in, if you want to show pronunciation. Or if you're writing Braille music, you need to be assured that when you write out, for example, a TH sign which is a crotchet C, that it will be written out without the computer trying to translate that into a TH sign. We need a Braille first format.

This is where BRF comes in.

So, within that, there are two uses for it. We've stored the Braille. The two uses for a Braille file beyond storing Braille are quite simply to emboss it out in hard copy or to read it on an electronic Braille display.

Let's talk about how a BRF file is constructed. At its core, it's a plain text file. It's the same sort of file that you would create in an application like Notepad on a computer. It's not the same sort of thing as a Microsoft Word document. This is really the difference between BRF and PEF. PEF would have been more like a Microsoft Word document because you can put bold and italics and you can put styles in and things like that. A BRF file is not. It's very straightforward. It's the sort of file that you would load into Notepad.

The way this works is that there are 64 unique combinations. There are the 63 dot patterns plus the space. Each of these dot patterns have been assigned a key on the keyboard or a character, if you like. The way these assignments were drawn up is not random, although it might seem random until you get used to how they work.

Again, I'm not quite sure what came first. I'm not sure whether BRF files came first or the US computer Braille code came first, but there is this thing called the US Computer Braille Code and it has the symbol assignments that are most commonly used in BRF files.

The other use for the US Computer Braille Code, apart from to write BRF files very quickly, is if you're for example programming and you need that one to one character mapping to make sure that code lines up or something, then you can use the US Computer Braille Code for that, or indeed the UK Computer Braille Code and much more talk about that is beyond the scope of this session but you just need to know that BRF files predominantly are based on the US Computer Braille Code.

So, the letters A to Z look like the letters A to Z. There's nothing surprising about that. If I wanted to write the word "Hello" in a BRF file, I would literally write the word "H-E-L-L-O", no problems there.

However, the lower signs, how to write the lower signs? By the lower signs, I mean dot 2, a lower b, middle C, lower d, lower e, lower f, all those sorts of things, lower g, lower i, lower j. To write those we use the numbers. So, for example, if I wanted to write a full stop, I would write the number four. So, if I wanted to write "Hello, full stop," I would write, "H-E-L-L-O 4." If I wanted to write "Hello, exclamation mark," I would write, "H-E-L-L-O 6." And if I wanted to write "Hello, question mark," I would write "H-E-L-L-O 8."

In practice you don't need to worry about this too much because you'll be either translating a BRF file using Duxbury or some other Braille translation program or you'll be writing in six-key entry on something like an Orbit Reader and you won't need to worry it because the Orbit Reader will put in the right symbols at the right time. So this is just so you have some idea of how we constructed all of this.

Similarly there are random combinations of dots, like, for example, the dot 6 capital. Well, the dot 6 capital, because we can write a comma as a number one, we don't need the comma to be dot 2, so we can use the comma as dot 6. So if I wanted to write a capital "HELLO, full stop," I'd write "comma H-E-L-L-O 4."

And that's how we do it. There's basically a symbol for every character in Braille. If you want the "of" sign, you do a left parenthesis. If you want a "with" sign, you do a right parenthesis. If you want an "and" sign, use the ampersand, and so it goes on.

There is a handout that will be made available when the recording is made available and for people who are interested, there is actually a whole list of all of the 64 Braille characters and what symbol you would use for each one, if you want to write BRF by hand, but, as I say, you probably don't need to in most cases.

BRF files that originate in countries where English is not the native language, and I won't concentrate too much on this, but it's not strictly necessary to use the US Computer Braille Code. Most BRF files do. If we were writing French in an English context, so if we were an English transcriber writing a French text book, we would still use the US Computer Braille Code and for the e acute, we would just use the symbol for the "for" sign.

If you're in France, they might use the French Computer Braille Code. There's nothing wrong with that. You just need to tell your Braille device that your Braille file was written in French Computer Braille Code instead of English Computer Braille Code. Honestly, don't worry about it. Assume that it's written in US, unless it doesn't render properly.

The other important characteristic to note about a BRF file is its layout. The layout is very strict. There's usually a reason why a BRF file was created. It was either created for reading on a Braille display or it was created for embossing. It was rarely created for both. If a BRF file was created for embossing, it might, for example, have page numbers. It might have running headers. It might have contents pages, all of this sort of thing. These have to be mapped out very, very precisely.

If, for example, your page is 40 characters long and you want a page number to be in the last two cells of the line, the only way to achieve this, because it's only a plain text format, is literally to do 38 spaces and then the number sign and then the letter i if it's page nine or what have you.

Similarly, if you want to centre something on that line, if you want to centre twelve colons on that line, you'd have to do 14 spaces and then the 12 colons which would be 3 3 3 3 3 3 3 3 3 3 3 3. Or indeed in UEB, quote 3 3 3 3 3 3 3 3 3 3 3 3, for the dot 5 that you would need in front of those colons.

This means that the layouts need to match for best results. If you have a BRF file that was written for a Perkins page, so it's 40 characters to the line, and you want to emboss that BRF file on an A4 page, good luck to you. It's very difficult to do it because what will happen is because it was encoded for a 40 character line, and your A4 page only has, say, 32 characters on that line, the embosser will try to break the line at the appropriate point. So, what you'll get is line of 32 cells and then on the next line, the remaining eight cells that are overspilled. It will not play nice with pagination. You'll probably only get page numbers every two pages or what have you, because it's introducing an extra line all the time and it will generally do quite nasty things. The answer to this problem, buy some Perkins paper if you've got a file that needs Perkins paper.

If you're in reverse, if you've got an A4 file and all you've got access to is Perkins paper, that's fine, you'll just get a really, really wide right margin.

The same sorts of things happen on Braille displays unless you've got certain settings turned on, and this is why peculiarities happen on the Orbit Reader when, for example, sometimes you get a hyphen at the end of a Braille line, for word division, and then you find that the Orbit has just sort of carried on but it's put a space after the hyphen and you wonder why there's a random space after the hyphen when there shouldn't be.

1 What's happened is that the Orbit Reader has intelligently worked out that you don't actually want to go onto the next line, so it's just added a space and carried on. But if that setting is turned off, what you would find on the Orbit Reader is you'd have probably two long lines followed by one short line or something like that.

On a BrailleNote, if you're reading a 38 cell to the line BRF file on a 32 cell BrailleNote, you'll quite often end up in a situation where you've got a full line and then probably about a fifth or a sixth of a line because the BrailleNote has had to take a new line and then eight cells later, the BRF file has also taken a new line. So, it's important to bear that in mind. That said, this is just stuff that you get used to. If you read BRF files on electronic Braille displays on a regular basis, you get used to this and it stops bothering you but that's why it happens.

The other really important thing is that it's hard to change Braille codes. So, let's take a step backwards. If your native language is English, but you've been sent a Microsoft Word document in French, the document will always be in French. If you want that document to be in English, you've got to use something like Google Translate, other translations are available. You've got to use a tool like that. You can't just go into Microsoft Word and change your settings and there you go, the document's in English now.

It's the same, because BRF is a Braille first format, it's only storing the contractions that the transcriber wants it to store. So, if you have a BRF file, and that BRF file is in UEB Grade 2 and you would like to read that in UEB Grade 1, you're out of luck. You cannot go into your settings and change the settings to Grade 1 and by magic, the BRF file will update. The BRF file stores the Braille information in Grade 2.

If you want to read it in Grade 1, and this is why we have problems with the SD card on the Orbit Reader, because not all books are in Grade 1, you need a separate BRF file.

If you're feeling adventurous, you can create one. You can use some translation software, you can back-translate the Grade 2 file and then forward-translate it in Grade 1. You're welcome to try. Your mileage will vary. Some BRF files, that works amazingly well, others, especially if they've got tables or they've got contents pages or page numbers, be prepared for a lot of that formatting to get lost and the pagination to just go all over the place. It is possible to do it and if you spend enough time editing the file, then you'll get decent results out of it.

So, that has been a very brief overview of what BRF files are and what they're not and what you can do with them. I will stop for questions in just a moment. Before I do that, I want to talk about where you get BRF files from. The easiest answer to that question is from your library, from your specialist blindness library. So, for example, RNIB Reading Services has a collection of BRF files or indeed BRF files are on the card that you get with your Orbit Reader.

BARD, in the States, Braille and Audio Reading Downloads, from the National Library Service, also has BRF files. The Centre for Equitable Library Access in Canada, that is CELA, I believe that has BRF files. And bookshare.org, which is predominantly in the US but it does have an international flavour, it also has BRF files.

As I've already mentioned, you can create your own BRF files, if you want to. To do that, you'd need a Braille translation program, like Duxbury or Braille2000 or BrailleBlaster or indeed the free tool Send To Braille which doesn't make particularly nice looking BRF files but it does make BRF files. We covered Send To Braille in our Markdown episode which is on our Media page on Taking Notes For Others.

If there's time, I'll run a quick demo of the Send To Braille tool because it's quite a useful tool for creating quick and dirty BRF files from, say, Microsoft Word or PDF or TXT or what have you, but you can do this fairly easily.

In a second, I'm going to talk about how to read a BRF file. I'm going to talk about how to navigate them and how to emboss them. I'm actually going to stop to get a drink of water now and while I do that, if people want to ask any questions, we'll probably take questions for about 15 minutes or so.

Ben Mustill-Rose: Thank you, Matthew, a great, thorough introduction there. As Matthew said, we'll take questions. If you'd like to ask a question, there are a couple of ways you can do so, depending on what device you're using but they all involve raising your hand. If you're on Windows, you can press Alt-Y to raise your hand, if you're on a Mac, you can press Option-Y. If you're dialling on a telephone, you can press star-nine. If you're an iDevice such as an iPhone, iPad or iPod Touch, you can press the More button which is in the bottom right of your screen and I believe that's visually represented by three dots and then once you're pressed that, you'll find a Raise Hand button.

We try to give people a little bit of warning before we go to them, so for example the first person we're going to go to is Bart and we say that just in case anyone's making a cup of tea, just so that we don't catch up them out. So, we're going to come to Bart first and then Steve.

Bart: I'm not an English speaker so I have problems with the English contractions and so my question is, why would I want to read a BRF file on my Braille display? There are, first of all, the problems with the line length that you explained. If I have a 32 cell display and my file is for 40 characters, it's not ideal. But why not use the Braille table, transcription table in the screen reader? If that's a good Braille table, it should give me more or less the same results on my Braille display as I would have had embossed on paper. So, why use this formatted document?

And also the disadvantage of reading BRF is that there is no real support of the synthetic voice. You explained you use a list of characters to represent Braille but if you let it speak by synthetic voice, there is no good support for that. So, it would really be a disadvantage for me to read such a book, because you cannot turn off the Grade 2 and my question would also be, why not make the books in Unicode Braille because these are, I think, a bit more standard?

Matthew Horspool: Those are brilliant questions and I'm glad you asked them, because in all of that spiel, it was one of the things I neglected to mention. Why wouldn't you just use the built-in translation? There's absolutely no reason actually. If you have built-in translation and it works for you, go ahead and use that. That said, there are situations where the built-in translation might not work quite as you expect. So, for example, if I'm using a Braille device and the first one that springs to mind is the Brailliant BI 40X, it's an amazing machine and I really like it and I'm not trying to say anything negative about it, but from my experience of it, if you're loading a book into it, in, say, Microsoft Word, or what have you, it's not the best at dealing with the bold, italics, underlining signs. So, if you're reading a book with lots of italics in it, and you need those italics for some reason, at present, a human transcribed book would be a better option for you.

Similarly the ongoing problems in England of quotation marks versus apostrophes, a human transcriber is likely to have sorted those problems out, whereas an automated system might not have done, and there are various little anomalies, if you like, where you might find that the translator is not quite doing the job that you want it to do and actually a BRF file would be better.

Similarly for tables, if there's a table, the transcriber will have done a nice job at transcribing that table, hopefully, and it might render slightly better and transcriber's notes for illustrations and other things, such as specialist codes like maths and music, there are various reasons why a BRF file might be the better choice.

You are right, it doesn't support synthetic speech very well at all, because of all the random symbols.

The other advantage is that they're very small and some devices don't have built-in translation at all, so the Orbit Reader, the basic Orbit Reader, you can't use a Word document on the Orbit Reader, it doesn't work. This means that the Orbit Reader loads books amazingly quickly. I have never seen a device load a book as quickly as the Orbit Reader and literally a book could be a couple of hundred kilobytes. It's tiny, small, absolutely wonderful.

That kind of leads me on to why you don't use Unicode Braille. This has to do with the size of Unicode characters but my understanding is, if you used Unicode Braille, you would essentially double the size of the BRF file. This might not be a problem on modern machines but when BRF was established, apart from the fact that Unicode Braille didn't exist back then, size was very much of the essence.

I hope that's answered the questions.

Ben Mustill-Rose: Thanks for that, Bart, from Braille Authority Belgium there. Great questions. We're going to come to Steve Pulley next and then Claire.

Steve: I've got a couple of questions. One is actually on the Braille Unicode again. I just wondered how that is stored, if you happen to know. Is it a number for each dot or is it a bit for each dot, as in a byte? The second way of doing that would keep things somewhat shorter but I'm just vaguely curious.

My main question really was about the formatting in the BRF files. Are there other characters which are used to help with the formatting, like, for example, carriage returns, line feeds, form feeds for a page and stuff like that? The reason why I'm asking is I've got some BRF files which I wouldn't mind putting out to an embosser but the last time I did that, it kept losing the second half of the line. It was meant to be a 40 character line, 25 lines per page, but it seemed to be losing the lines. I don't know whether that's because the embosser was expecting to see new line characters in whatever format.

Matthew Horspool: Quite possibly, and I don't want to get into specifics about what embosser were you using and what translator and what software because we could be here all day trying to debug it, but there are problems like that. I'll talk a bit more about that later in the session and special characters that you can use for navigation, because you are right. You could navigate, for example, by print page number or by the lines of colons that occur between chapters and things. So, there are definitely ways to do it and I'll come onto those a bit later, if you don't mind.

Steve: I don't mind at all. As far as you know, do embossers tend to use their own particular way of interpreting things like new lines?

Matthew Horspool: Embossers have a really unique character. Embossers are amazing machines and the best thing I can say, you sound like you know what you're talking about, so the best thing you can do is make friends with your embosser and then your embosser will make friends with you and it will do exactly what you want it to do, but you have to know how to get the best out of your embosser and you are right, different embossers do it in different ways. I'm very familiar with the Index range of Braille embossers and Index, if it sees a line break, it will take a line break, no problem. If it sees a form feed, it will take one. There were problems where, for example, if you had a 29 line BRF file and your Index was set to 29 lines, Index would eject the paper because it would think it had reached the end of the page, and then it would see a form feed and then it would eject again because it didn't take into account the fact that it had already ejected.

So, there's all sorts of little quirks on embossers and part of it also depends on the software and the way the BRF file was created and you get long lines and short lines and dots missing and all sorts of things, if you're not careful. The trick is to make sure that the embosser and the BRF file match exactly or the embosser exceeds it. So, if you're embossing a 40 by 25, if you set your embosser to 26 lines, that's normally fine. It will normally cope with that and it will eject when it gets to 25 and it'll go onto the next page. But if you set it to 24 lines, it won't. It will go onto the next page for line 25 and then it'll go onto the next page again. Similarly if you set it to 41 or 42 cells, it's normally fine, if you set it to 40, it's normally fine, but if you set it to 38, it'll go onto the next line for two characters and then drop onto the next line again when it sees the line break.

Get to know your embosser and I love this sort of thing, I could spend all day talking about the relationship between embossers and software and all this sort of thing. I would be curious to know what embosser you're using but perhaps by email, because I really could literally talk about this all day and never move on to the next section. Drop a line to help@braillists.org and I'd love to investigate this further.

On the Unicode blocks, I'll be very quick because I don't have the answer in full, but essentially Unicode's based on 8 dot Braille, so there are 128 Unicode characters in a block and each one of those characters represents a combination. So there's a dot 1 character, a dot 1 2 character, etc., so I think it's one bit per dot but James Bowden might have more information and we'll try to get that to you by email because it's getting quite technical.

Ben Mustill-Rose: Thanks for that, Steve. We're going to come to Claire next and then, Matthew, I notice that we do have George Bell with his hand up with what I assume is a comment as opposed to a question. So, we'll go to Claire first.

Claire: It took me ten minutes to get into the site to listen to you and so I missed the beginning and I'm not very clear what a BRF file is.

Matthew Horspool: It's a file where it's Braille first, so for example a print file, you have letters and numbers, a BRF file is a Braille file and it's basically you can specify that the file had a dot 6 and then it had an M and then it had an A and then it had a T and then it had a THE contraction and then it had a W. So, it's a way of representing Braille dots, as opposed to print letters and numbers.

Claire: I'm a bit lost, I'm afraid.

Matthew Horspool: There will be a recording of the session and what I'll do is I'll try to remember to email you but if you go to braillists.org/media probably early next week, the recording will be posted there and perhaps it'll go a bit slower and you'll be able to stop and rewind and maybe get a better idea.

Ben Mustill-Rose: So, Matthew, more questions or content?

Matthew Horspool: I've just noticed we've got James and George. It might be worth taking both of those comments and then moving on because James will have an interesting point on Unicode.

Ben Mustill-Rose: So we'll go to George first and then James.

George: BRF, maybe we're out in the clouds somewhere, but I always understood it to stand for Braille Ready Format, not File.

Matthew Horspool: That's good to know and I'll update the notes.

George: I'm not sure if I misheard you or not but I think you said something about BRFs having formatting information.

Matthew Horspool: No, BRFs don't have formatting information. Any formatting you do want, you'd have to put it in with spaces and new lines and form feeds.

George: Finally, we do get a lot of support calls along the lines of what you were talking about with embossers. I have fought and am winning that the dog should wag the tail, not the tail wag the dog. In other words, you set your embosser up to its maximum defaults and allow your software to control the embosser. So, it doesn't matter if you want to send, in the most recent cases, a 47, 48 cells per line or 30 characters per line. The embosser will be told what to do and thankfully that's becoming more and more helpful these days and certainly reduces our support efforts.

Matthew Horspool: Thanks, George. That's kind of unique to Duxbury at this point although the Index drivers and other embossers do have the facility to do it. So, if other Braille translators want to implement it, the spec is out there.

James Bowden: Steve, you were asking about the Unicode Braille. This is going to be technical, by definition. It's Unicode 28XX, from 2800 to 28FF and it is a bit per dot. So 2800 is no dots or Braille space, 2801 is Braille dot 1, 2802 is Braille dot 2, 2804 is Braille dot 3, 2808 is Braille dot 4 and so on. So, the six dot Braille code will use Unicode 2800 to 283F and then eight dot Braille code will add to 2840 up to 28FF. That is very technical but I hope that answers your question.

Ben Mustill-Rose: Well, there you are. Where else could you find out this stuff? Thank you, James.

Matthew Horspool: Thank you. I'll carry on and go through this next bit as fast as I can, without going so fast that people can't understand what I'm saying.

I'd like to start with the practicalities. How do you read a BRF file? If you've been given a BRF file, say, you've downloaded one from Reading Services or BARD or Bookshare or any of the others, how do you actually read it?

Well, the good news is, if you've got an Orbit Reader, you don't need to do anything, apart from copy the BRF file that you've downloaded to the SD card and I'm not going to demonstrate that because we've only got 15 minutes left, but essentially you can plug the Orbit Reader into your computer and then it shows up like a drive, like a USB stick, and you can copy the file to the Orbit Reader and open the BRF file on the Orbit Reader and it will read and it will be in the grade of Braille that you want, hopefully, or at least the grade of Braille that you expected and no further action is required.

That's the case on quite a few Braille displays actually, that have these sorts of functionalities. You should just be able to open it up, either in the reader app or in some cases in the editor program.

If you've got a Braille display connected to a computer, so I used to have the previous generation, Brailliant BI 40, rather than the current BI 40X, and the thing about the previous generation, BI 40, was that you couldn't copy files to it. You literally had it connected to a screen reader and that was all you had. You could connect it to a smartphone or a tablet but it couldn't work in a standalone mode.

So what happens if I want to read a BRF file in that sort of way? Well, you can do it. There are various ways to do it. You could open the BRF file in Duxbury and there's information about how to do that in the handout. If you don't have a copy of Duxbury, you could just open the BRF file in Notepad which I'm going to do. I've found a BRF file and I'm going to press Enter on it and hope that you can hear my speech.

JAWS: Arrow ueb update course brf 40 zx20 7 l dot brf notepad computer braille.

Matthew Horspool: And it's opened in Notepad and I'll maximise the window and if I down arrow...

JAWS: Blank update six u f apostrophe seven uk dollar i and seven.

Matthew Horspool: ...you can hear that I have lots of nonsense that the speech is saying. On the Braille display, I also have a whole bunch of nonsense. Now, exactly how you fix this will depend on your screen reader. I happen to be using JAWS. I'm not going to do a step by step for every single screen reader under the sun but I will quickly demonstrate it in JAWS so that you can see the sort of setting that you might be looking for in another screen reader.

Basically what you want to do is turn the translator off and use computer Braille. So, if I go into the JAWS settings centre with JAWS key and six, which I have done and there seems to be a bug with JAWS at the moment where it crashes, so we'll just wait patiently until JAWS turns itself back on again.

What I'll do, once it turns itself back on again, is I will go into the tree view and look for Braille and then look for the Braille table settings and change those settings.

Here we go, it's come up. I'm not going to go the default file because I only want computer Braille to be on for Notepad.

JAWS: Zero user closed one of braille closed.

Matthew Horspool: So I'll go to Braille.

JAWS: Braille open si-- one general closed general open two translation closed one of ni-- translation opened three language english united kingdom one of six translation general braille.

Matthew Horspool: Okay, so language English United Kingdom, that's fine.

JAWS: Output computer braille.

Matthew Horspool: So, you'll notice we've got computer Braille. If I press the space bar on here--

JAWS: Q dot K english grade 1 Q dot K english grade 2 unified english braille grade 1 unified english braille grade 2 computer braille one of five.

Matthew Horspool: So, it's on computer Braille. Computer Braille is actually the one that I want. There's another option in here that I need to tell you about.

JAWS: Show current word in computer braille and-- suppress capital signs and omit computer braille tables dot dot dot two braille mode structure two of nine and enclosed active cursor follow braille cursor follows display text in eight dot braille mode.

Matthew Horspool: That's the one I want. Display text in eight dot Braille mode. The reason why this is important is because a number of BRF files are encoded with all of their letters in capitals. There's no real harm in this except that if you are using eight dot Braille, every cell virtually will have a dot 7 underneath it and that is really uncomfortable to read.

So, you turn off the eight dot Braille mode in JAWS or in NVDA, there's an equivalent option. In most screen readers, there's an equivalent option. So, if you turn off the eight dot Braille mode and use six dot or failing that, if you have a choice of US eight dot computer Braille or US six dot computer Braille, if you choose US six dot computer Braille, then that will get rid of all of those annoying dot 7s that appear all of the time.

Similarly on the Orbit Reader, there's a setting called "Filter Dot 7". If you toggle that setting on, then you'll find that the Orbit Reader doesn't show dot 7s all the same. It's on by default, so if it's working for you, you don't need to change anything.

If you're navigating a BRF file, the thing to know is that there's no mark-up in these files so you need to have some idea of what the file looks like and you need to have some idea of what you're looking for. You can't just press a button and say, "Next chapter" or "Previous chapter" or what have you. On some machines, you might be able to press next page or previous page, but really you need to be able to navigate the BRF file by knowing what's around.

So, for example, you could use the page up and page down key or you could use the previous and next paragraph key, or you could use the Find facility. The Find facility is this amazing thing in the world of BRF because it will literally allow you to find anything. So, for example, I'm at the top of this...

JAWS: Number A.

Matthew Horspool: ...BRF file which has a number A on it, because it's page one. If I do a control-F--

JAWS: Find, find what, octopus all--

Matthew Horspool: And I was looking at Octopus Energy earlier on which is why that's come up. I'm going to type 33333. On a Braille keyboard, I would type dots 2 5, five times. So, five middle Cs. And I'm going to press Enter.

JAWS: Computer braille.

Matthew Horspool: And nothing's happened but actually something has happened. If I escape, I've got the line and if I move to the next line, it says, "In one volume." So, I'll press the F3 key to move to the next one.

I could keep doing this--

JAWS: Blank blank blank blank 333 billion 300 blank blank blank blank.

Matthew Horspool: Okay.

JAWS: 300 number b k a d.

Matthew Horspool: So, 2014.

JAWS: 300 blank blank 3 blank blank 3 blank.

Matthew Horspool: The reason why this is taking such a long time is because there's a whole load of stuff here. If I move a bit further down and try it.

JAWS: 333 billion blank blank a number b eight number b four m k s number b 4 up to n m k s.

Matthew Horspool: Okay, so this is reading nonsense but on the Braille display, I can see that I've moved to section two, punctuation marks, brackets seven. If I press F3 again and move down...

JAWS: Blank blank page break number number z four.

Matthew Horspool: ...I can see I'm now on number three, capitalisation. So, searching for things like lines is very helpful, "3 3 3" or middle C, middle C, middle C. Searching for things like dot 5, middle C, and a number sign, if you want to find a particular print page number, there are various techniques like that. There's a whole list of them in the handouts. They all basically rely on you using the Find facility of your screen reader or your application or what have you, and just finding things.

If you get stuck, the best thing you can do is go to the top of the file and then Find, because that way you know that you're searching through the whole file, whereas if you're in the middle of the file and you do a Find, you'd have to remember to find forwards and backwards.

The final thing I'd like to talk about very briefly is embossing a BRF file and I won't touch on Duxbury because we've only got five minutes but there is a section in the handout to do with it. If you're embossing a BRF file and you're not using Duxbury, you need to know some information about the BRF file. You need to know how many characters per line it's got, and how many lines per page it's got.

So, I have this BRF file open in Notepad and I can find this information out very easily. If I got to the top of the file...

JAWS: Top number a.

Matthew Horspool: ...it says number A, that's fine. That's a page number. So, if I press the End key, and now I press the command to read the status line which in JAWS is insert page down...

JAWS: L n one call 41 100 percent windows z r l e f 8.

Matthew Horspool: I don't need to worry about most of that but it said, "Ln one call 41." That's actually because I'm on the carriage return so we can subtract one from that and we get 40 cells to the line and actually I could just count now.

JAWS: Number 8 blank update blank blank blank blank 3 in 3 blank blank blank blank number uk for number three blank blank blank blank blank f slash for 333 bill-- number b k blank page break num--

Matthew Horspool: So, I counted 27 lines and then I heard the magic words...

JAWS: Page break number.

Matthew Horspool: ...page break. So there were 27 lines to the page break so that means there are 27 lines to the page and indeed I can verify this if I press the status bar key now.

JAWS: L n 28 call one 100 percent windows z r.

Matthew Horspool: Line 28, so I'm on the first line of the second page which means that the first page finished on line 27. So, this particular file has 27 lines to the page and 40 characters to the line and as long as I tell my embosser that that is the case, my embosser will emboss that file accurately.

If I don't tell my embosser that that is the case, as I was explaining to Steve earlier on, you might have problems.

If you're in Duxbury, there's a check box called "Import BRF File Without Interpretation." That's a good check box to know about. If you don't check that check box, you need to know about all the other options in Duxbury and there's information about that in the handout.

With only four minutes to go, I shall turn it back over for some more questions.

Ben Mustill-Rose: Thank you, Matthew. I'll give people a little bit of time to raise their hands. We should have time for a few. As a quick reminder, you can find a copy of this recording when it goes up and a copy of the recordings of all of our previous events, along with handouts and increasingly transcriptions of the narration, by visiting our Media page at www.braillists.org/media. Someone called it a treasure trove of resources the other day which I was quite proud of.

So there are a couple of hands. We're going to come to Lisa and then Kawal.

Lisa: Thank you so much for this information sharing. You had mentioned that if your BRF file is in UEB Grade 2 and you want to read it in Grade 1, that you'll need a separate BRF file for Grade 1 and then also you had pointed out sources of BRF files, some of the online libraries like NLS, bookshare.org, and I've noticed that in browsing some of the online libraries, when you look at the different file types that are available and you can choose to download the BRF file, there's often no information up front as to what type of BRF file it's in, what grade, what format and I was just wondering, is there a way to know up front what version you're getting?

Matthew Horspool: Yes, so first of all, libraries need to get better at giving us that metadata and some libraries are better than others. RNIB is pretty good. I know others are not. Once you've opened the file, sometimes it's immediately obvious, you'll open it and you'll go, "Yes, that's definitely UEB," or "That's definitely SEB." So my best advice is look for something that's really obvious in SEB that's not in UEB or something really obvious in UEB that's not in SEB. For example, look for the contraction "to the." If you do a Find on a lower f followed by a THE sign, and you press Enter, that's a six exclamation mark, if you're on a computer and you want to be technical, if you find something like that, then that's an SEB book because UEB doesn't have that contraction. Or, for example, if you search for dot 5 and the GH sign, that's a bracket in UEB. If it doesn't find it, it's not foolproof that it's not UEB but if it does find it, it's pretty much foolproof that it is UEB. It's just about picking some signs that are pretty obviously SEB or UEB signs and that's the only way I've found of being able to tell.

Ben Mustill-Rose: We're going to finish off with a question from Kawal.

Kawal: Firstly, I've never done a BRF file in Notepad and second of all, I want to know if you wanted to write something in Notepad, would you use computer Braille and if you wanted to save something, what type would you use in the Save As box?

Matthew Horspool: You probably would never write a BRF file in Notepad unless you have lots of time on your hands. I actually do write BRF files in Notepad. When I was at Exhall, I used to have to write French, English documents, English vocabulary lists or French vocabulary lists and that's another great use for a BRF file, by the way, because of all the language switching. But I literally used to sit there with the QWERTY keyboard and type equals signs and exclamation marks for the accented letters and do it that way. But that is what you'd have to do. There isn't really a way of doing six key entry in Notepad unless you've got a Braille display, in which case you could use the Braille keyboard, or you could use something like Perky Duck. You don't necessarily have to use Notepad. You could do it in Perky if you wanted to do it or just in Duxbury.

Assuming that you've written out the BRF file in Notepad, when you go to Save, you would save as All Files, and then you would just add dot brf to the end of the file name and that would make sure it saves as a BRF.

Ben Mustill-Rose: Wow, manual writing of BRFs in Notepad, sounds like a bit of a lockdown kind of activity. Great questions from everyone. I know I've learned a huge amount in this session and once again, many thanks to Matthew for hosting. I'll pass things over to Dave to wrap up.

Dave Williams: Thanks, Ben, and if you found some of the information in tonight's session a little bit technical, please don't panic. More help is available. You can of course contact us, help@braillists.org. You can also join the Braillists forum from our website at braillists.org and a recording, as Matthew said, of this session will be made available at braillists.org/media. That's probably going to come up next week and will also be available as part of Braillecast from wherever you normally receive your podcasts.

Thank you very much, Matthew, for a very informative session, quite dense session, I would say. We've got another, I suppose, technical session next time but very different and that's Braille On Android and that's our topic in two weeks' time.

This time next week we'll be having an open forum and letting people ask any Braille-related questions in our Braille Bar, so do bring your questions, maybe you have some follow-up questions from tonight that you might like to ask next Tuesday evening from 7:30pm, UK time.

Don't forget the Braille book club on Thursday at 6pm and our newsletter.

Thank you to Ben for your help with the moderation and everyone else for your expertise and fantastic questions.

From all of us at the Braillists Foundation, have a great week and we'll speak with you very soon. Bye for now.