As we all know, several extended characters are not handled properly by Score when a non-35 font is used. There are at least 15 concerned characters as ©, ñ, ™, etc. Of course there are walk-arounds, but as I had to do a work with some of these (ã, õ, etc) for a composer who wanted to include them as "special sounding voyells" in a piece for voice and chamber orchestra (one of these masterworks written by an yet unknown genius which - and who - will probably soon vanish in the fog of History), I decided to take this opportunity and try to see how it could be set correctly once for all with the request fonts.

When using the 35 fonts with FM00.PSC to FM34.PSC, it works. So obviously the problem comes from the FMxx.PSC files as made by FontConv.exe from AFM files, and I'm convinced that the .PSCs which come with Score, though maybe built up with FontConv, have been edited by hand by LS before packing them - I hardly imagine that LS could have provided a different version of FontConv that the one he used himself. Therfore the idea is it must be possible to edit those FontConv-made .PSC in order to correct the wrong values. I'm far to be an expert of computers, I know nothing serious about all these bulk- and dump- or whatever programs, so I had to do it with simple tools and methods. As it leaded to something, I thought it would be useful for other users.

AFM files are in text format (easy to handle) and look like this:

```
StartFontMetrics 2.0
Comment(s)
FontName
FullName
FamilyName
Weight
(...)
StartCharMetrics xxx (= number of described characters)
C 32 ; WX 278 ; N space ; B 0 0 0 0 ;
C 33 ; WX 333 ; N exclam ; B 110 -14 222 677 ;
(...)
C -1 ; WX 500 ; N mu ; B 28 -216 497 447 ;
(...)
EndCharMetrics
StartKernData             (This optional kerning section
StartKernPairs xxx       is not used by FontConv)
KPX ...
(...)EndKernPairs
EndKernData
StartComposites           (This optional section seems to have
EndComposites            definitively disappeared from AFM files)
EndFontMetrics
```

The C field is the character's number as encoded in the font; when it says -1 it means that the character is not built-in but will be drawn on the fly when necessary ("combined" letters as diacriticals, which merge an encoded plain letter plus an encoded accent, etc). The WX field is the width of the character (what means X? I do not know, as for KPX). The N field is the name of character as defined by the PostScript fonts conventions. The B field is something I cannot remember exactly, probably something as a bounding box to be used when necessary.

So I started studiying the article published about the question by Jan de Kloe on his site, entitled "PSC files, FONTCONV and SCORE character width" (Jan de Kloe, August 2002) and I'm totally indebted to his text for getting started, as it gave me a unvaluable hand to put mine (hands) in the dirt. In this article one may find, a description of the structure of the file and the method of encoding the various width values that FontConv reads from the AFM file and writes to the FMxx.PSC file. However, with all respect due to Jan, I went to slightly different conclusions about some points.

First, the structure seems to me to have a difference with Jan's description. Though I may be wrong, IMHO here is how it goes:

```
byte   | dec. value
--------------------------------
1        75          File header (probably for Score to identify it)
--------------------------------
2        129        Section header telling how many bytes
                       follow, including itself

3-30                 Font name, completed with spaces or
```

		truncated up to 28 characters (read from the line "FontName" in the AFM
31-130		100 bytes to be read by couples, which give 50 integers. Each couple a-b makes a value of [a + b*256].
131	129	Section trailer with same value as previous section header
132	129	Section header (same as above)
133-260		128 bytes --> 64 integers
261	129	Section trailer (same as above)
262	98	Section header
263-360		98 bytes --> 49 integers
361	98	Section trailer
362	129	Section header
363-490		128 bytes --> 64 integers
491	129	Section trailer
492	129	Section header
493-620		128 bytes --> 64 integers
621	129	Section trailer
622	70	Section header
623-692		70 bytes --> 35 integers
693	70	Section trailer
694	130	File trailer

Two anomalies are noticeable : in the section starting at byte 262 there are 98 bytes. Therefore the value of byte 262 should be 99 (including itself). But it's 98 (and the section trailer is 98 too). Same remark for section starting at byte 622, which should has a value of 71 but has 70 instead (sim. for the trailer). As Jan suggested directly to me, Score probably does not really reads these headers / trailers, it just knows that there are present and skips them.

As a result, every FMxx.PSC file is 694 bytes long. If it's not, Score immediately crashes, in its elegant usual way of freezing everything to dead.

The inside reason of such a complicated structure is not clearly known, but Jan discussed it in this article which you may read if interested. By all means, every header and trailer have always these values and are always in the same place. It's easy to show by building a fake AFM file with a fixed similar value for each character and feeding FontConv with it.

A second point of unagreement with Jan's article is about the B field. Jan states that

> The B-field is not of interest to us as it is not used by FONTCONV.

In fact it is used, and we'll see that a bit later.

Each integer a-b holds a width information. The next step was to find which character's width was asssigned to which a-b couple. Jan spots out that FontConv has a table which "pipes" each value to the right place, and tells us where it is in the FontConv executable. (That's the kind of thing I'm totally unable to discover by myself.) Now, when looking at the table, stranges things appear: many characters are indeed drove into a specific socket, while others are switched to the same one, 163. What does it means? I found later that this 163 socket matches font's character 247 which is commonly "undefined". Jan wrote

> If someone is interested in the translation routines I have developed for the PSC interpretation, just ask (source only, VB6, includes the equivalence table).

so I took advantage of this and asked. Jan very kindly sent it to me, and I found that his routine gave the same result as my results. So once again, I (well, a little Basic routine) made another fake AFM file in which each WX field was set to the same value as the C field, plus 1000 (in order to leave Fontconv reading values which could make sense instead of taking a risk of crashing, one never knows with these delicate little things). The result was quite efficient and provided a shortcut to avoid playing with FontConv inside tables and all that didling.

Now we see what PS character each a-b refers to. But we see other things.

Some a-b bytes do not refer to a character: they have a value of 1. As the minimum should have been 1000 (1032, actually), it proves that for some reason FontConv made no calculation at all on these bytes, it just set them to 1. Checking which ones is of highful interest. When comparing FM00.PSC as packed with Score and an oustide FMxx.PSC made by FontConv, we see that some a-b bytes are set to 1 in both files, while others have a valuable width in FM00.PSC but are set to 1 in the outsider PSC. Obvioulsy these ones refer to our missing extended characters. But we may still learn something else. I made a complete cook with three "real" AFMs: one from a standard Adobe font (with original AFM), another from the font Lausanne, used by Jan (see his article) and kindly provided by Kr. Rogalski (made by Fontographer), and the last one from a Monotype Times-Roman font (AFM built from PFB by GhostScript). The three output PSCs (translated in decimal) have been aligned in regard of FM00.PSC. We may see that some a-b bytes always get a value of 1 for every font; some other a-b have a value of 1 for each font but FM00.PSC (my guess, as said, is that LS edited these ones by hand in FM00 to FM34); and some other a-b have a value of 1 in one font but got a width value in another. In this last case I understand that FontConv was ready to make the calculation, but depending of the font, failed, probably because the AFM has not the proper information. In the case of having 1 for any font, it shows that the socket remains empty for ever, and as you may see there is a good amount of place remaining free...

Finally I managed to find out which a-b was referring to which missing character. You'll find hereafter a complete table, giving the a-b couple, the matching character, eventually the typing to get it with Score, and the width value assigned by FontConv for each of four fonts I tried, as examples.

However, something remains unclear for me. Each C xx character has its socket in the PSC, and a good lot of C -1 have theirs too, as Ñ (Ntilde) or å (aring). But some rather common have not, as é (eacute) or ç (ccedilla). However they are handled correctly by every font, 35 or not-35. These are invoked by the << etc processes (v. ?x or !x), and are displayed by Score with special luxe (the accent or umlaut is on screen, due to a special displaying routine), though by themselves these special strings just call an assigned character in FontInit.PSC, exactly as for ?x or !x etc. Where are they in the PSC? I do not know.

Now, we found that the assignment to characters go only up to byte 360. Byte 361 (as I see it) is the section trailer, byte 362 is the next section header, and from there bytes get a value which has nothing to do with widths, they come from the B fields of the AFM. Again, a fake AFM shows this: I made one with changing all four values of B into easily reckognized ones, and as a result we see that FontConv picks up the third number of the B field as the a-b value. What's the use of it? I just wonder if it has really one for Score. As a trial, I made a PSC in which all bytes from 363 had a value of 0 (except headers and trailers), and input Code16s referring to it. Score displayed and printed them without a sneeze. Either I missed understanding what it is and had luck not to get Score crashed, or it's something LS planned to use but finally let down without upgrading FontConv. Having found that, I did not spend time to find which characters match which position in this section, but if really necessary it could be done by another fake AFM.

Anyway, the last step here is to edit the wrong values in the PSC to get what we want. By chance every wrong character has got a specific a-b, so all we have to do is to pick up the width in the AFM, to make a quick calculation in order to get the a-b and to update the wrong 1 into their right values in the PSC. No doubt the next issue could be to develop a special program for this, but in fact it can be done by each of us as I did it myself, not with clever tools or techniques (I do not know them), but quite easily with, for instance, Edit (Microsoft's) which we all have.

There I am now. As it is my work is uncomplete, but I have not enough knowledge to go on. So I would be very happy and interested if some experts would tell me about the lacking points:

- how are handled the <<, >>, %% and co? Are they in PSCs, or is there a secret trick?

- what's the use of the second half of the PSC file, from a-b 363-364 on?

- what happens to characters piped to "163" (FontConv table), i.e. a-b 359-360?

Hereafter is the table.

Chanvrelin

****

(Widths)

Bytes	character	(typed in Score by)	Score's Times	MTimes	AGaram	Lausanne
31-32	0		500	500	500	556
33-34	1		500	500	500	556
35-36	2		500	500	500	556
37-38	3		500	500	500	556
39-40	4		500	500	500	556
41-42	5		500	500	500	556
43-44	6		500	500	500	556
45-46	7		500	500	500	556
47-48	8		500	500	500	556
49-50	9		500	500	500	556
51-52	A		722	722	623	667
53-54	B		667	667	605	667
55-56	C		667	667	696	722
57-58	D		722	722	780	722
59-60	E		611	611	584	667
61-62	F		556	556	538	611
63-64	G		722	722	747	778
65-66	H		722	722	806	722
67-68	I		333	333	338	278
69-70	J		389	389	345	500
71-72	K		722	722	675	667
73-74	L		611	611	553	556
75-76	M		889	889	912	833
77-78	N		722	722	783	722
79-80	O		722	722	795	778
81-82	P		556	556	549	667
83-84	Q		722	722	795	778
85-86	R		667	667	645	722
87-88	S		556	556	489	667
89-90	T		611	611	660	611
91-92	U		722	722	746	722
93-94	V		722	722	676	667
95-96	W		944	944	960	944
97-98	X		722	722	643	667
99-100	Y		722	722	574	667
101-102	Z		611	611	641	611
103-104	.		250	250	250	278
105-106	,		250	250	250	278
107-108	(		333	333	320	333
109-110	)		333	333	320	333
111-112	a		444	444	404	556
113-114	b		500	500	500	556
115-116	c		444	444	400	500
117-118	d		500	500	509	556
119-120	e		444	444	396	556
121-122	f		333	333	290	278
123-124	g		500	500	446	556
125-126	h		500	500	515	556
127-128	i		278	278	257	222
129-130	j		278	278	253	222
131	[129]					
132	[129]					
133-134	k		500	500	482	500

```
135-136 l                         278 278 247 222
137-138 m                       778 778 787 833
139-140 n                        500 500 525 556
141-142 o                        500 500 486 556
143-144 p                        500 500 507 556
145-146 q                        500 500 497 556
147-148 r                        333 333 332 333
149-150 s                        389 389 323 500
151-152 t                        278 278 307 278
153-154 u                        500 500 512 556
155-156 v                        500 500 432 500
157-158 w                        722 722 660 722
159-160 x                        500 500 432 500
161-162 y                        500 500 438 500
163-164 z                        444 444 377 500
165-166 :                        278 278 250 278
167-168 ;                        278 278 250 278
169-170 ?                        444 444 321 556
171-172 !                         333 333 220 278
173-174 +                        564 564 500 584
175-176 -                         333 333 320 333
177-178 *                         500 500 394 389
179-180 /                          278 278 327 278
181-182 =                         564 564 500 584
183-184 _ (underscore)          500 500 500 556
185-186 ’ (quoteright)            333 333 235 191
187-188                            1 1 1 1
189-190 "                          408 408 404 355
191-192                            1 1 1 1
193-194                            1 1 1 1
195-196                            1 1 1 1
197-198                            1 1 1 1
199-200                            1 1 1 1
201-202                            1 1 1 1
203-204                            1 1 1 1
205-206                            1 1 1 1
207-208                            1 1 1 1
209-210                            1 1 1 1
211-212                            1 1 1 1
213-214                            1 1 1 1
215-216                            1 1 1 1
217-218                            1 1 1 1
219-220                            1 1 1 1
221-222 (sp)                     250 250 250 278
223-224 ã  ~a                   444 1 1 1
225-226 Ã  ~A                   722 1 1 1
227-228 ñ  ~n                   500 1 1 1
229-230 Ñ  ~N                   722 1 1 1
231-232 õ  ~o                   500 1 1 1
233-234 Õ  ~O                  722 1 1 1
235-236 • (bullet) !0           350 350 388 278
237-238 „ (quotedblbase !1    444 444 384 556
239-240 ” (quotedblright) !2   444 444 404 500
241-242 ¡ (exclamdown) !3     333 333 220 333
243-244 ¢  !4                     500 500 500 333
245-246 £  !5                     500 500 500 556
247-248 §  !6                     500 500 506 556
249-250 ¤  !7                     500 500 500 333
251-252 ' (quotesingle) !8       180 180 235 737
253-254 “ (quotedblleft) !9      444 444 404 667
255-256 Å  !A                     722 1 1 1
257-258 ‡  !D                     500 500 480 222
259-260 Š  !S                      556 1 1 1

261   [129]
262   [98]

263-264 Zcaron  !Z                 611 1 1 1
265-266 å  !a                         444 1 1 1
267-268 †  !d                         500 500 480 333
269-270 … (ellipsis) !e            1000 1000 1000 556
```

```
271-272 ƒ  !f                            500 500 500 260
273-274 «  !g                           500 500 378 556
275-276 »  !h                           500 500 378 556
277-278 fi  !i                            556 556 522 737
279-280 ‹ (guilsinglleft) !j           333 333 233 584
281-282 › (guilsinglright) !k         333 333 233 584
283-284 fl  !l                            556 556 522 611
285-286 - (emdash) !m               1000 1000 1000 722
287-288 - (endash) !n                 500 500 500 584
289-290 ¶  !p                            453 453 500 537
291-292 ¿  !q                            444 444 321 500
293-294 š  !s                            389 1 1 1
295-296 ¥  !y                            500 500 500 667
297-298 zcaron !z                       444 444 377 1
299-300 #                                500 500 500 556
301-302 $                                 500 500 500 556
303-304 %                                833 833 844 889
305-306 &                                 778 778 818 667
307-308 ‘ (quoteleft) (\\)              333 333 235 333
309-310 - ?-                              564 1 500 1
311-312 < (less)                        564 564 500 584
313-314 > (greater)                    564 564 500 584
315-316 @                                  921 921 765 1015
317-318 Æ  ?A                            889 889 850 556
319-320 Œ  ?E                            889 889 1012 556
321-322 L-slash  ?L                     611 611 559 500
323-324 Ø  ?O                            722 722 795 556
325-326 [   ?[                             333 333 320 278
327-328 \   ?\                             278 278 309 278
329-330 ]   ?]                             333 333 320 278
331-332 æ  ?a                             667 667 585 556
333-334 ©  ?c                             760 1 1 1
335-336 œ  ?e                             722 722 729 556
337-338 dotless i (\\)                 278 278 257 556
339-340 l-slash  ?l                     278 278 261 333
341-342 ø ?o                               500 500 486 556
343-344 ® ?r                               760 1 1 1
345-346 ß ?s                               500 500 523 556
347-348 ™ ?t                               980 980 1100 1
349-350 {  ?{                              480 480 320 334
351-352 |   ?|                                200 200 239 260
353-354 }   ?}                              480 480 320 334
355-356 ª (ordfeminine) ?f             276 276 332 556
357-358 º (ordmasculine) ?m          310 310 387 556
359-360  (undefined)                     333 333 360 584

361   [98]
362   [129]


               (Third B values)

363-364   476 464 457 519
365-366   394 378 383 359
367-368   475 458 464 507
369-370   431 417 446 522
371-372   472 465 467 523
373-374   438 434 430 514
375-376   468 461 468 518
377-378   449 455 479 523
379-380   442 442 443 517
381-382   460 457 451 514
383-384   706 712 643 654
385-386   596 613 558 627
387-388   637 632 676 681
389-390   689 685 734 674
391-392   597 588 574 616
393-394   544 517 492 583
395-396   704 709 712 704
397-398   703 703 766 646
399-400   316 307 300 188
```

401-402	376	384	305	489
403-404	709	733	698	663
405-406	598	588	553	537
407-408	871	870	877	761
409-410	709	710	755	646
411-412	688	685	749	739
413-414	542	524	517	622
415-416	701	685	885	739
417-418	654	677	680	684
419-420	491	504	437	620
421-422	594	583	650	597
423-424	705	712	738	644
425-426	701	711	704	647
427-428	936	934	994	928
429-430	706	712	671	648
431-432	703	711	611	652
433-434	597	581	628	588
435-436	183	179	181	191
437-438	202	199	192	191
439-440	304	311	300	299
441-442	284	292	240	265
443-444	442	440	412	530
445-446	474	466	465	517
447-448	412	411	398	477
449-450	491	505	502	499
451-452	421	412	379	516
453-454	383	434	404	262
455-456	470	482	453	499
457-458	490	496	499	491
459-460	259	253	235	155
461-462	212	192	182	155
463-464	500	506	501	501
465-466	259	253	228	155
467-468	764	774	768	769
469-470	490	496	504	491
471-472	470	466	451	521
473-474	470	464	472	517
475-476	498	506	494	494
477-478	335	340	330	332
479-480	348	356	293	464
481-482	279	278	300	257
483-484	479	500	502	489
485-486	468	491	444	492
487-488	694	714	677	709
489-490	479	488	434	490

491 [129]
492 [129]

493-494	476	494	459	489
495-496	418	426	375	469
497-498	196	193	181	191
499-500	202	213	192	191
501-502	395	402	276	492
503-504	224	222	166	187
505-506	534	545	443	545
507-508	289	293	288	289
509-510	437	429	358	349
511-512	302	278	297	295
513-514	534	545	443	545
515-516	500	508	500	556
517-518	242	241	180	132
519-520	1 1 1 1			
521-522	337	343	325	285
523-524	1 1 1 1			
525-526	1 1 1 1			
527-528	1 1 1 1			
529-530	1 1 1 1			
531-532	1 1 1 1			
533-534	1 1 1 1			
535-536	1 1 1 1			

537-538	1	1	1	1
539-540	1	1	1	1
541-542	1	1	1	1
543-544	1	1	1	1
545-546	1	1	1	1
547-548	1	1	1	1
549-550	1	1	1	1
551-552	1	1	1	1
553-554	0	0	0	0
555-556	442	1	1	1
557-558	706	1	1	1
559-560	490	1	1	1
561-562	709	1	1	1
563-564	470	1	1	1
565-566	688	1	1	1
567-568	300	298	333	202
569-570	417	413	339	547
571-572	417	413	349	464
573-574	224	223	166	311
575-576	448	440	445	321
577-578	491	477	494	537
579-580	426	424	456	512
581-582	503	487	467	293
583-584	133	139	161	752
585-586	399	413	350	620
587-588	706	1	1	1
589-590	439	432	429	242
591-592	491	1	1	1
593-594	597	1	1	1
595-596	442	1	1	1
597-598	440	451	430	277
599-600	891	889	889	537
601-602	490	499	507	167
603-604	449	469	333	459
605-606	468	469	333	459
607-608	521	543	500	752
609-610	271	276	188	545
611-612	288	276	188	545
613-614	521	543	503	588
615-616	1007	1009	1000	674
617-618	507	509	500	545
619-620	373	457	463	497

621	[129]
622	[78]

623-624	395	401	257	469
625-626	351	1	1	1
627-628	502	499	514	685
629-630	418	426	375	1
631-632	495	482	491	529
633-634	456	448	444	520
635-636	772	799	799	850
637-638	750	746	805	645
639-640	230	241	181	211
641-642	534	1	443	1
643-644	536	545	445	536
645-646	536	545	444	536
647-648	819	898	735	868
649-650	869	866	840	530
651-652	877	866	1002	516
653-654	598	588	559	477
655-656	688	685	749	516
657-658	299	297	301	250
659-660	361	278	279	295
661-662	245	251	203	215
663-664	634	635	568	491
665-666	717	1	1	1
667-668	690	691	712	489
669-670	259	253	235	528
671-672	259	271	285	332

673-674	470 481 451 489
675-676	718 1 1 1
677-678	468 466 496 518
679-680	945 966 1038 1
681-682	341 376 285 292
683-684	132 120 149 167
685-686	370 376 257 292
687-688	278 278 330 530
689-690	301 297 355 516
691-692	323 304 301 333

693 [70]
694 [130]

****