

Title: The Cyrillic TimeScore family of fonts
Author: Jan de Kloe
Date: July 4th, 2003
Version: 1.02 – updated January 11, 2004

Introduction

When Score users need Cyrillic characters for modern Russian, they can use the family of fonts named TimeScore. The family consists of four members: TimeScore, TimeScore-Bold, TimeScore-Italic, TimeScore-BoldItalic and there is the font TimeScoreAcc for some special characters. The set was made available by Sergey Lebedev of the Moscow Conservatory. He allowed me to put the collection on my website for downloading. This document describes the font capabilities and instructions for use. First the alphabets are handled, then at the end the details of TimeScoreAcc with capabilities for CE (Central European) characters are explained.

In this article, Cyrillic characters are displayed in **Arial Bold** to easily distinguish them from Latin script. Since the Score set has some characters which cannot be visualized in MS-Word (eg. **ю** with acute), these characters are described.

A prerequisite for understanding this article is the knowledge of Score Code16 items and printer loading operations, while a minimum of Russian helps.

There are other Cyrillic Type 1 fonts available for Score users but these have different numeric values. They will be described in a different document. Languages using Cyrillic script with non-Russian characters such as pre-Revolutionary Russian, Macedonian, etc. are not supported in TimeScore. The modern Russian set also covers Bulgarian and Mongolian.

The font

TimeScore character set	
Lowercase Russian alphabet	абвгдежзийклмнопрстуфхцчшщъыьэюя ё
Uppercase Russian alphabet	АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ Ё
Digits	0123456789
Non-alphabetic characters	. , () : ; ? ! + - * = № " [] { } – (not all shown)

The Bold, Italic, and Bold-Italic variations are not shown.

How to use the font

The set is based on phonetic or visual resemblance with the Latin alphabet which makes it easy to input characters without a Cyrillic keyboard. For instance to type Russian **АБЦ** you type ABC. The full set of characters is available in the demo file CYRTEST.MUS which accompanies the set. A more complicated character would be **Щ** which corresponds to m-dash which in Score is !m.

These are the 'one key' equivalents:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	\$	&
а	б	ц	д	е	ф	г	х	и	й	к	л	м	н	о	п	ч	р	с	т	у	в	ж	я	ы	з	э	ё
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	@	%
А	Б	Ц	Д	Е	Ф	Г	Х	И	Й	К	Л	М	Н	О	П	Ч	Р	С	Т	У	В	Ж	Я	Ы	З	Э	Ё

Here is the table for letters which cannot be obtained with one alphabetic key:

?E	!g	!h	?o	?t	%%e
ш	щ	ь	ъ	ю	ё
!p	!m	?O	!q	!s	%%E
Ш	Щ	Ь	Ъ	Ю	Ё

This shows that the characters **ё** and **Ё** can be made in two different ways.

Then for ethnomusicological purposes, professor Lebedev included the accented vowels which can be entered with these combinations:

<<a	<<e	<<i	<<o	<<u	?a	!0	!1	!2	!3
a with acute	e with acute	и with acute	o with acute	y with acute	ы with acute	э with acute	ю with acute	я with acute	ё with acute
<<A	<<E	<<I	<<O	<<U	?A	!4	!5	!6	!7
A with acute	E with acute	I with acute	O with acute	Y with acute	Ы with acute	Э with acute	Ю with acute	Я with acute	Ё with acute

The TimeScore fonts also contain a number of special characters:

description	keyboard	graphic	description	keyboard	graphic
exclamation	!(¹)	!	question mark	?(¹)	?
double high quote	"	"	brackets	?[?]	[]
number	#	№	braces	?{ ?}	{ }
left and right parentheses	()	()	backslash	?\	\
asterisk	*	*	circumflex	\\330 ⁽³⁾	^
plus	+	+	underscore	\\331 ⁽³⁾	_
comma	,	,	vertical line	?	
hyphen	-	-	tilde	\\332 ⁽³⁾	~
period	.	.	single quote	!8	'
slash	/ or \	/	n-dash	!n	-
digits	0 – 9	0 - 9	grave	\\301	`
colon	:	:	breve	\\333 ⁽³⁾	˘ ⁽²⁾⁽⁴⁾
semicolon	;	;	bullet	\\334 ⁽³⁾	• ⁽²⁾⁽⁴⁾
less	<	<	acute	\\335 ⁽³⁾	ˆ ⁽²⁾⁽⁴⁾
equal	=	=	question mark	?(¹)	?
greater	>	> ⁽²⁾			

- (1) when followed by space or at end of Code16 item
- (2) not in TimeScore
- (3) cannot be made in Score with default FONTINIT.PSC
- (4) not in TimeScore-Bold

Uncommon characters

There is no way in Score to input characters such as the underscore, or those for which there is no equivalent code assigned. The flexibility of Score however is such that any character which is hardwired or softloaded in a font can be equated to an existing Score character whether it has a character representation ('a') or is in the non-display category ('?L', '!5', '~n', or \331). The elements we need for this are:

- the numeric value which Score generates (ie. zero for 'á');
- the fact whether a character in a font is encoded by AFM and carried over to PSC;
- the possible re-assignment of a character by FONTINIT.PSC which is copied to the printed output.

We do not have the AFM files for the TimeScore family of fonts but the information is also available in PFB from which I extracted which characters exist and which of those are not encoded¹. The function of AFM and FONTINIT.PSC is explained further down.

In order to make existing encoded and not encoded font characters available to Score, one must adapt FONTINIT.PSC. Since there are some of those special situations in our family of fonts, we show here what is needed to get them:

	circumflex ^	underscore _	tilde ~	breve ˘	bullet •	acute ´
TimeScore, TimeScore-Bold	\330	\331	\332	(1)	(1)	(1)
TimeScore-Italic, TimeScore-BoldItalic	\330	\331	\332	\333	\334	\335

⁽¹⁾ In these fonts, these characters are not present.

If you need the above characters, the following change to FONTINIT.PSC is necessary for which you do not need to sacrifice any other character:

```
/acc [ 0/aacute 1/agrave 2/acircumflex 3/adieresis
4/ccedilla 5/atilde 6/minus 7/aring 8/eacute 9/egrave
10/ecircumflex 11/edieresis 12/iacute 13/igrave 14/icircumflex
15/idieresis 16/oacute 17/ograve 18/ocircumflex 19/odieresis
20/otilde 21/uacute 22/ugrave 23/ucircumflex 24/udieresis
25/ydieresis 26/ntilde 27/scaron 28/zcaron 29/copyright
128/Aacute 129/Agrave 130/Acircumflex 131/Adieresis 132/Ccedilla
133/Atilde 134/registered 135/Aring 136/Eacute 137/Egrave
138/Ecircumflex 139/Edieresis 140/Iacute 141/Igrave 142/Icircumflex
143/Idieresis 144/Oacute 145/Ograve 146/Ocircumflex 147/Odieresis
148/Otilde 149/Uacute 150/Ugrave 151/Ucircumflex 152/Udieresis
153/Ydieresis 154/Ntilde 155/Scaron 156/Zcaron 157/trademark
8#362/onequarter 243/onehalf 244/threequarters 252/underscore
8#330/asciicircum 8#331/underscore 8#332/asciitilde 8#333/Scaron
8#334/scaron 8#335/zcaron
245/onesuperior 246/twosuperior 247/threesuperior ] def
```

¹ Not encoded means that a character exists in the font but has no numeric value. It was defined in AFM by the C -1 command

Only two lines were inserted before the last line and it would be easy to copy this portion of text into your FONTINIT.PSC. There is still a drawback to this solution in that you can only address these characters as octal values in a text string (ie. as \332 to obtain a tilde). When octal values are present in a Code16 item, their width is not incorporated for justification and centering because the width of these characters is not defined in the corresponding PSC file.

Note that in the equate table above, some characters are redefined (scaron first as decimal 27, later as octal 334). Only the last definition is valid.

Problems and restrictions

Note that some keyboard characters are refused by Score. On my United Kingdom keyboard I have the ‘not’ character (¬) and the pound sign (£) which refuse to be entered. Specifying a character which exists in Score for which there is no equivalent in a font (such as ?L in a Cyrillic item) results in the character being ignored. Experiments with non-existing characters have sometimes led to a run time error in Score. Although the exact situation is not yet understood, the following message appeared: “Error writing to file or device. Press any key to continue”. The sending item number shown is probably one too low. This message was even produced when writing to an EPS file so it must be related to the Code16 item rather than to the file or device, and the file was not properly closed. When this happened during printer output, the printed needed to be restarted. Traditionally, the messages of Score are not of much help.

Installation of the fonts

To make the TimeScore family of fonts available on your system you need to unzip TIMESCOR.ZIP and move some files to your systems. Score works with font numbers and you can select these numbers in the range 35 – 89. The PSC files have been prepared for font numbers 50 – 54 and you can simply rename them if your Score MUS files use other prefixes for Cyrillic. For this document we assume the font numbers as they come. The set contains:

- PSC files FM50.PSC – FM54.PSC (to tell Score the width of characters). These need to be copied to your LIB directory;
- PFB files TIS____.PFB, TISI____.PFB, TISB____.PFB, TISBI____.PFB and TISA____.PFB (loadable fonts). These go to the directory where you keep loadable fonts. These are the files you need to send to the printer before printing Score files which use these fonts;
- PFM files (metrics information incase you use Acrobat Distiller to convert EPS files to PDF format).

Normally, Acrobat Distiller expects PFB files in the \Font directory (in my case C:\Program Files\Adobe\Acrobat 5.0\Resource\Font) and PFM files in \Font\PFM.

To check the installation, you can print the test file CYRTEST.MUS which refers to all fonts in the family. Sample output of this file can be seen in the PDF file.

The meaning of AFM files and the function of the Score file FONTINIT.PSC.

The inner workings of fonts in a Score system are further explained, but for every day use of Cyrillic, the following information is not required.

Table 1 at the end of the report was obtained from the PFB files which were loaded to a printer (with PCSEND or DownWind), then dumped with a PostScript™ program (SHOWFONT.PS)², and represented here in a format which resembles an AFM file. The AFM files for these fonts are not available and not necessary for Score users.

AFM (Adobe Font Metrics) files³ define details such as character width, PostScript language character name⁴, character bounding box sizes, activation of a character (encoded or not encoded), kerning details, etc. The PSC files which Score needs in order to know the width of characters are generated from AFM files by the Score utilities FONTCONV.EXE and FONTCVT4.EXE. Note that changing the width of a character in AFM and hence PSC only influences the way Score does justification, centering and JT – it does *not* influence the font itself so the characters will still print using the original width as it is in PFB.

² SHOWFONT.PS written by Dylan McNamee can be downloaded from ftp.cs.washington.edu

³ Adobe Font Metrics File Format Specification, Version 4.1, 7 October 1998 is downloadable from the Internet

⁴ These names such as ‘Agrave’ are stored within a hard or soft font

Normally, Acrobat Distiller expects PFB files in the \Font directory (in my case C:\Program Files\Adobe\Acrobat 5.0\Resource\Font) and PFM files in \Font\PFM.

There are four columns in table 1: The numeric value (C-field), the PostScript language character name (N-field), a graphic representation if there is one in my MS-Word, and comments.

As can be seen, there are slight differences between the Cyrillic fonts. When the C-field indicates -1, this means that the character exists in the font but that it is not encoded.

The file FONTINIT.PSC which comes with Score is there to equate characters which exist in Score but are not encoded in a font. Score includes FONTINIT.PSC in the EPS file once, when any non-display character (for EPS) occurs in a Code16 item. It assigns a character to an internal numeric Score value. For example, the Latin character 'á' which in Score is written as '<<a' generates in EPS code '\000' meaning that Score's numeric value for the letter 'a' with the acute diacritical. Here is how it would look in an EPS file:

```
/Times-Roman          f [ 771.442 0 0 771.442 0 0] mkf sf
 750 -25050 m save (\000) show
```

However, the numeric value for 'á' in a normal font such as TimesRoman is not assigned. The character however exists in the font with its symbolic name 'aacute'. In file FONTINIT.PSC, there is the equation of the value \000 to 'aacute' which tells the printer that when it encounters this \000, it should print 'á'. This is the line in that file containing the equate:

```
/acc [ 0/aacute 1/agrave 2/acircumflex 3/adieresis
```

Such an equation can be done to a non-existing character number like zero through 31, or 127, or to an existing one. In the latter case the character is replaced. Such equation is only valid for the duration of one print job (one EPS file). The printer does not remember the equates for a subsequent job.

Values of characters in FONTINIT.PSC are normally in decimal notation, so when you see '29/copyright' this means character 29 in Score made by ?c is to become the copyright sign. Score generates special characters as EPS codes in octal, and the copyright shows up there as \035. The Calculator in your Windows systems has a scientific 'view' which allows conversion between decimal and octal.

You can make changes to FONTINIT.PSC but you have to know the details of the font (all existing characters in the set with their symbolic name, which characters are not encoded or can be replaced) and the numeric values Score uses for them. The standard file which comes with Score is valid for most common fonts. You can use octal notation as there is one example in the line:

```
8#362/onequarter 243/onehalf 244/threequarters 252/underscore
```

which means that '242/onequarter' is the decimal equivalent of the first equate.

The FONTINIT.PSC as it comes is usable for Cyrillic since the following not encoded characters are equated as described. In standard Russian you do not need the letters with an acute but they are provided for ethnomusicology. However, this group contains also the common letters 'Ě', 'ě', and 'ю'. These are shown in Table 2.

There are however characters in the set which are not addressed by these equates. If we would want to use such characters, FONTINIT.PSC needs to be adapted as shown above.

TimeScoreAcc

There are only ten characters in this font shown here:

description	graphic	octal	decimal	Score
C hashek	Ć	103	64	C
c hashek	ć	143	99	c
c acute	č	144	100	d
breve	˘	306	198	\\306
dot accent	·	307	199	\\307
ring	°	312	202	\\312
hungarian umlaut	¨	315	205	\\315
ogonek	˛	316	206	\\316
caron	ˇ	317	207	\\317
dotless i	ı	365	245	\\365 ⁽¹⁾

⁽¹⁾ An additional definition in FONTINIT.PSC is required to replace the ‘one superior’ by this character used in Turkish. Replace the last two lines by:

```
246/twosuperior 247/threesuperior  
8#365/dotlessi ] def
```

such that the definition of ‘onesuperior’ now becomes the new character.

Note that all these characters can be made with standard Times-Roman and comparable fonts, and you will not really need this font. The Central European character set is subject of an other article.

The octal characters can be used as well and the table follows of the equivalences. Some duplicates occur and practically all characters shown can be made with existing Score codes. There will be hardly any need for octal addressing.

	0	1	2	3	4	5	6	7
000	á							
010	é			ë	и+			
020	ó					ý		
030								
040		!	”	№	э	Ё	ё	
050	()	*	+	,	-	.	/
060	0	1	2	3	4	5	6	7
070	8	9	:	;	<	=		?
100	Э	А	Б	Ц	Д	Е	Ф	Г
110	Х	И	Й	К	Л	М	Н	О
120	П	Ч	Р	С	Т	У	В	Ж
130	Я	Ы	З	[\]	^	
140		а	б	ц	д	е	ф	г
150	х	и	й	к	л	м	н	о
160	п	ч	р	с	т	у	в	ж
170	я	ы	з	{		}	~	
200	Á							
210	É			Ё	И+			
220	Ó					Ý		
230						Ю		
240		ё+	Э+	Ю+				Я+
250	Ё+	'		щ	ý			
260		—					Ш	э+
270		ю+	я+	ь				Ъ
300		`						
310	Ё			ё				
320	Щ							
330	^		~					
340		Ы+		Є				
350		Ь	Ш					
360		Ы+						
370		Ъ		Ю	_			

Acknowledgement

I am grateful to Sergey Lebedev for making these fonts available and for letting me put them on www.dekloe.be for downloading. They are used by SipXML, the converter of MusicXML to Score, and by SipText, the Score text editor with the Cyrillic extension.

Table 1

C 32 ; N space ;	space	
C 33 ; N exclam ;	!	
C 34 ; N quotedbl ;	"	
C 35 ; N numbersign ;	№	
C 36 ; N dollar ;	э	
C 37 ; N percent ;	€	
C 38 ; N ampersand ;	ö	
C 40 ; N parenleft ;	(
C 41 ; N parenright ;)	
C 42 ; N asterisk ;	*	
C 43 ; N plus ;	+	
C 44 ; N comma ;	,	
C 45 ; N hyphen ;	-	
C 46 ; N period ;	.	
C 47 ; N slash ;	/	
C 48 ; N zero ;	0	
C 49 ; N one ;	1	
C 50 ; N two ;	2	
C 51 ; N three ;	3	
C 52 ; N four ;	4	
C 53 ; N five ;	5	
C 54 ; N six ;	6	
C 55 ; N seven ;	7	
C 56 ; N eight ;	8	
C 57 ; N nine ;	9	
C 58 ; N colon ;	:	
C 59 ; N semicolon ;	;	
C 60 ; N less ;	<	
C 61 ; N equal ;	=	
C 62 ; N greater ;	>	not in TimeScore
C 63 ; N question ;	?	
C 64 ; N at ;	Э	
C 65 ; N A ;	А	
C 66 ; N B ;	Б	
C 67 ; N C ;	Ц	
C 68 ; N D ;	Д	
C 69 ; N E ;	Е	
C 70 ; N F ;	Ф	
C 71 ; N G ;	Г	
C 72 ; N H ;	Х	
C 73 ; N I ;	И	
C 74 ; N J ;	Й	
C 75 ; N K ;	К	
C 76 ; N L ;	Л	
C 77 ; N M ;	М	
C 78 ; N N ;	Н	
C 79 ; N O ;	О	

C 80 ; N P ;	П
C 81 ; N Q ;	Ч
C 82 ; N R ;	Р
C 83 ; N S ;	С
C 84 ; N T ;	Т
C 85 ; N U ;	У
C 86 ; N V ;	В
C 87 ; N W ;	Ж
C 88 ; N X ;	Я
C 89 ; N Y ;	Ы
C 90 ; N Z ;	З
C 91 ; N bracketleft ;	[
C 92 ; N backslash ;	\
C 93 ; N bracketright ;]
C 94 ; N asciicircum ;	^
C 95 ; N underscore ;	_
C 97 ; N a ;	а
C 98 ; N b ;	б
C 99 ; N c ;	ц
C 100 ; N d ;	д
C 101 ; N e ;	е
C 102 ; N f ;	ф
C 103 ; N g ;	г
C 104 ; N h ;	х
C 105 ; N i ;	и
C 106 ; N j ;	й
C 107 ; N k ;	к
C 108 ; N l ;	л
C 109 ; N m ;	м
C 110 ; N n ;	н
C 111 ; N o ;	о
C 112 ; N p ;	п
C 113 ; N q ;	ч
C 114 ; N r ;	р
C 115 ; N s ;	с
C 116 ; N t ;	т
C 117 ; N u ;	у
C 118 ; N v ;	в
C 119 ; N w ;	ж
C 120 ; N x ;	я
C 121 ; N y ;	ы
C 122 ; N z ;	з
C 123 ; N braceleft ;	{
C 124 ; N bar ;	
C 125 ; N braceright ;	}
C 126 ; N asciitilde ;	~
C 161 ; N exclamdown ;	ë with acute
C 162 ; N cent ;	Э with acute
C 163 ; N sterling ;	Ю with acute

C 167 ; N section ;	Я with acute	
C 168 ; N currency ;	Ě with acute	
C 169 ; N quotesingle ;	'	
C 171 ; N guillemotleft ;	Щ	
C 177 ; N endash ;	–	
C 182 ; N paragraph ;	Ш	
C 183 ; N bullet ;	э with acute	
C 185 ; N quotedblbase ;	ю with acute	
C 186 ; N quotedblright ;	я with acute	
C 187 ; N guillemotright ;	ь	
C 191 ; N questiondown ;	Ъ	
C 193 ; N grave ;	`	
C 200 ; N dieresis ;	Ë	
C 203 ; N cedilla ;	ë	
C 208 ; N emdash ;	Ц	
C 225 ; N ÆE ;	Ы with acute	
C 233 ; N Oslash ;	Ь	
C 234 ; N OE ;	Ш	
C 241 ; N ae ;	ы with acute	
C 249 ; N oslash ;	ъ	
C 251 ; N germandbls ;	Ю	
C -1 ; N Aacute ;	Á	
C -1 ; N Eacute ;	Ě	
C -1 ; N Edieresis ;	Ë	duplicate of percent
C -1 ; N Iacute ;	И with acute	
C -1 ; N Oacute ;	Ó	
C -1 ; N Scaron ;	ˇ	<i>in TimeScore-Italic and TimeScore-BoldItalic</i>
C -1 ; N Uacute ;	У with acute	
C -1 ; N Ydieresis ;	•	<i>in TimeScore-Italic and TimeScore-BoldItalic</i>
C -1 ; N aacute ;	á	
C -1 ; N eacute ;	é	
C -1 ; N edieresis ;	ë	duplicate of ampersand
C -1 ; N iacute ;	и with acute	
C -1 ; N nbsp ;	non-breaking space	not in TimeScore
C -1 ; N oacute ;	ó	
C -1 ; N scaron ;	•	<i>in TimeScore-Italic and TimeScore-BoldItalic</i>
C -1 ; N trademark ;	ю	
C -1 ; N uacute ;	у with acute	
C -1 ; N zcaron ;	‘	<i>in TimeScore-Italic and TimeScore-BoldItalic</i>

Table 2

PostScript name	character	remark	Equated to (oct/dec)	Score
Aacute	Á		200/128	<<A
Eacute	É		210/136	<<E
Edieresis	Ë	duplicate of percent	213/139	%E
Iacute	И with acute		214/140	<<I
Oacute	Ó		220/144	<<O
Scaron	ˇ (breve)	<i>in TimeScore-Italic and TimeScore-BoldItalic</i>	233/155	
Uacute	Ÿ with acute		225/149	<<U
Ydieresis	• (bullet)	<i>in TimeScore-Italic and TimeScore-BoldItalic</i>	231/153	
aacute	á		0/0	<<a
eacute	é		10/8	<<e
edieresis	ë	duplicate of ampersand	13/11	%e
iacute	и with acute		14/12	<<i
nbspace	non-breaking space	not in TimeScore		
oacute	ó		20/16	<<o
scaron	• (bullet)	<i>in TimeScore-Italic and TimeScore-BoldItalic</i>	33/27	
trademark	Ю		235/157	?t
uacute	у with acute		25/21	<<u
zcaron	´ (acute)	<i>in TimeScore-Italic and TimeScore-BoldItalic</i>	34/28	