

File created: 17-Apr-2026 08:58:07 {MEDLEY}<sources>MCCSFONTS.;145

edit by: rmk

changes to: (VARS MATHTOMCCS SYMBOLTOMCCS SIGMATOMCCS)
(FNS MCCSMAPPAIRS)

previous date: 15-Apr-2026 22:12:41 {WMEDLEY}<sources>MCCSFONTS.;134

Read Table: INTERLISP

Package: INTERLISP

Format: UTF-8

(RPAQQ MCCSFONTSOMS

[(VARS NSFONTFAMILIES ALTOFONTFAMILIES)

(INITVARS MCCSFONTFAMILIES)

(FNS DISPLAYENCODINGFN)

(COMS

; Mapping to MCCS

(FNS MCCSCODEMAPARRAY MCCSMAPFN MCCSMAPPAIRS XCCS.CS0.UNDEFINED XCCSUNDEFINEDPAIRS)
(VARS ALTOTEXTTOMCCS SYMBOLTOMCCS SIGMATOMCCS HIPOTOMCCS CYRILLICTOMCCS MATHTOMCCS
PALATINOTOMCCS TITANTOMCCS TITANLEGALTOMCCS)
(GLOBALVARS GACHATOMCCSARRAY ALTOTOMCCSARRAY SYMBOLTOMCCSARRAY HIPOTOMCCSARRAY
CYRILLICTOMCCSARRAY MATHTOMCCSARRAY SIGMATOMCCSARRAY PALATINOTOMCCSARRAY
TITANTOMCCSARRAY TITANLEGALTOMCCSARRAY)

;; For translation of codes in datastructures (e.g. Tedit)

[INITVARS [GACHATOMCCSARRAY (MCCSCODEMAPARRAY (XCCSUNDEFINEDPAIRS '((Lowline ↑X)
(ALTOTOMCCSARRAY (MCCSCODEMAPARRAY ALTOTEXTTOMCCS))
(SYMBOLTOMCCSARRAY (MCCSCODEMAPARRAY SYMBOLTOMCCS))
(HIPOTOMCCSARRAY (MCCSCODEMAPARRAY HIPOTOMCCS))
(CYRILLICTOMCCSARRAY (MCCSCODEMAPARRAY CYRILLICTOMCCS))
(MATHTOMCCSARRAY (MCCSCODEMAPARRAY MATHTOMCCS))
(SIGMATOMCCSARRAY (MCCSCODEMAPARRAY SIGMATOMCCS))
(PALATINOTOMCCSARRAY (MCCSCODEMAPARRAY PALATINOTOMCCS))
(TITANTOMCCSARRAY (MCCSCODEMAPARRAY (XCCSUNDEFINEDPAIRS TITANTOMCCS)))
(TITANLEGALTOMCCSARRAY (MCCSCODEMAPARRAY (XCCSUNDEFINEDPAIRS TITANLEGALTOMCCS))

(COMS

;; Mappings into MCCS: needed for e.g. Tedit coercion. \TEDIT.MCCS.TRANSLATE

(FNS GACHATOMCODE SYMBOLTOMCODE SIGMATOMCODE ATOMCODE MATHTOMCODE HIPOTOMCODE
CYRILLICTOMCODE PALATINOTOMCODE TITANTOMCODE TITANLEGALTOMCODE))

[INITVARS [DISPLAYFONTCOERCIONS '(((HELVETICA (<= * 2))
(HELVETICA 4))
((MODERN (<= 15 * 16))
(* 14))
((MODERN (<= 17 * 21))
(* 18))
((MODERN (<= 22 * 28))
(* 24))
((MODERN (<= 29 * 33))
(* 30))
((MODERN (<= 34 * 40))
(* 36))
((MODERN (<= 41 * 65))
(* 48))
((MODERN (<= 66 *))
(* 72))
((PALATINO 9)
(PALATINO 12))
((PALATINO (<= * 8))
(PALATINO 10))
(TITAN (<= * 9)
BOLD)
(MODERN 10))
(TITAN (<= * 9)
ITALIC)
(MODERN 10))
(TITAN (<= * 9))
(TITAN 10))
(LPT AMTEX]

(DISPLAYCHARCOERCIONS '((GACHA TERMINAL)
(MODERN CLASSIC)
(TIMESROMAN CLASSIC)
(HELVETICA MODERN)
(TERMINAL MODERN)
(HIPPO CLASSIC)
(CYRILLIC CLASSIC)
(MATH CLASSIC)
(SIGMA MODERN)
(SYMBOL MODERN)
(TITAN CLASSIC)
(PALATINO CLASSIC)
(OPTIMA MODERN)
(BOLDPS CLASSIC)
(PCTERMINAL CLASSIC)
(TITANLEGAL CLASSIC]

;; Defunct coercions? Mapping for DOS filenames, Adobe equivalences

```

(COMS (INITVARS [ADOBEDISPLAYFONTCOERCIONS '(((HELVETICABLACK 16)
(HELVICABLACK 18))
(SYMBOL)
(ADOBESYMBOL))
(SYMBOL 11)
(ADOBESYMBOL 10))
(AVANTGARDE-DEMI)
(AVANTGARDE))
(AVANTGARDE-BOOK)
(AVANTGARDE))
(NEWCENTURYSCHLBK)
(CENTURYSCHOOLBOOK))
(BOOKMAN-LIGHT)
(BOOKMAN))
(BOOKMAN-DEMI)
(BOOKMAN))
(HELVETICA-NARROW)
(HELVETICANARROW))
(HELVETICA 24)
(ADOBEHELVETICA 24]

```

```

(*DISPLAY-FONT-NAME-MAP* '((TIMESROMAN . TR)
(HELVETICA . HV)
(TIMESROMAND . TD)
(HELVETICAD . HD)
(MODERN . MD)
(CLASSIC . CL)
(GACHA . GC)
(TITAN . TI)
(LETTERGOTHIC . LG)
(BOLDPS . BP)
(TERMINAL . TM)
(CLASSICTHIN . CT)
(HIPPO . HP)
(LOGO . LG)
(MATH . MA)
(OLDENGLISH . OE)
(SYMBOL . SY])

```

(RPAQQ NSFONTFAMILIES (CLASSIC CLASSICTHIN MODERN TERMINAL OPTIMA BOLDPS PCTERMINAL))

(RPAQQ ALTOFONTFAMILIES (TIMESROMAN TIMESROMAND HELVETICA HELVETICAD CLARITY BRAVOX TONTO CREAM OLDENGLISH))

(RPAQ? MCCSFONTFAMILIES NIL)

(DEFINEQ

DISPLAYENCODINGFN

[LAMBDA (FONTSPEC)

; Edited 8-Mar-2026 22:46 by rmk

(LET ((FAMILY (fetch (FONTSPEC FSFAMILY) of FONTSPEC)))

(if (MEMB FAMILY MCCSFONTFAMILIES)

then 'MCCS

elseif (MEMB FAMILY NSFONTFAMILIES)

then 'XCCS\$

elseif (MEMB FAMILY ALTOFONTFAMILIES)

then 'ALTOTEXT

elseif (EQ FAMILY 'TITAN)

then

;; The other sizes seem to have already been converted

(CL:IF (EQ 14 (fetch (FONTSPEC FSSIZE) of FONTSPEC))

'TITAN

'XCCS\$)

else FAMILY])

)

;; Mapping to MCCS

(DEFINEQ

(MCCSCODEMAPARRAY

[LAMBDA (MAP INVERT)

; Edited 8-Mar-2026 00:42 by rmk

; Edited 5-Feb-2026 11:02 by rmk

; Edited 2-Feb-2026 23:11 by rmk

; Edited 6-Sep-2025 18:26 by rmk

; Edited 31-Aug-2025 16:15 by rmk

; Edited 7-Aug-2025 08:55 by rmk
; Edited 2-Jun-2025 11:45 by rmk
; Edited 1-Jun-2025 07:26 by rmk
; Edited 24-May-2025 12:22 by rmk
; Edited 21-Dec-2024 18:57 by rmk

;; MAP is a list of (destcode source) pairs where source is a code, a (code font) pair, NIL, or (NIL), where the NIL sources designate a slug.

;; If INVERT, this produces a hash table for font-recoding by MOVEFONTCHARS where each key is a destcode the corresponding value is its source. The NIL source is represented as (NIL) in the hash table (as if it is (NIL font)), because putting NIL as a hash value takes it out of the MAPHASH. The INVERT hashtable presumably is used only offline and only once per font, when it is recoded to MCCS.

;; If not INVERT, this produces an array indexed by source codes that maps CS0 sources to their dest codes. This is used for translating old font-dependent codes in data structures (like Tedit) into the MCCS codes at the same time that the font is also changed to an MCCS font. This is used online but presumably only once per document/datastructure, if the result of the code translation is remembered.

;; The restriction to CS0 is because only XCCS fonts have characters outside that, and that's where MCCS agrees with XCCS. Other legacy fonts (HIPPO, SYMBOL...) only have CS0 glyphs. If a CS0 source code does not exist in the mapping, it is assumed to be an identity.

;;

```

(SETQ MAP (for PAIR SOURCE DEST in MAP when (AND (LISTP PAIR)
                                                  (NEQ '* (CAR PAIR))))
  collect (SETQ DEST (CAR PAIR))
          (CL:UNLESS (CHARCODEP DEST) ; DEST always designates a code
                    (SETQ DEST (CHARCODE.DECODE DEST)))
          (SETQ SOURCE (CADR PAIR))
          ;; NIL source is a slug
          (if (NULL SOURCE)
              then (SETQ SOURCE (CONS NIL))
              elseif (LISTP SOURCE)
                    then (CL:UNLESS (OR (CHARCODEP (CAR SOURCE))
                                         (NULL (CAR SOURCE)))
                                (SETQ SOURCE (LIST (CHARCODE.DECODE (CAR SOURCE))
                                                  (CADR SOURCE))))
                    elseif (CHARCODEP SOURCE)
                          else (SETQ SOURCE (CHARCODE.DECODE SOURCE)))
              (LIST DEST SOURCE)) ; Add identities for CS 0
  (for DEST from 0 to \MAXTHINCHAR unless (ASSOC DEST MAP) do (PUSH MAP (LIST DEST DEST))) ; Just easier to debug
  (LET (VAL)
      [if INVERT
        then ;; KEY's are destination codes, good for maphash recoding
          (SETQ VAL (HASHARRAY (LENGTH MAP)))
          (for PAIR in MAP do (PUTHASH (CAR PAIR)
                                       (CADR PAIR)
                                       VAL))
        else (SETQ VAL (ARRAY (ADD1 \MAXTHINCHAR)
                              NIL NIL 0 0))
          (for PAIR SOURCE in MAP eachtime [SETQ SOURCE (CAR (MKLIST (CADR PAIR)
                                                                      ; Value for NIL sources is NIL
                                                                      NIL))]
            when (CHARCODEP SOURCE) do (SETA VAL SOURCE (CAR PAIR)
                                              VAL)]
  VAL])

```

**(MCCSMAPFN
[LAMBDA (FROMENCODING)**

; Edited 5-Oct-2025 19:56 by rmk
; Edited 6-Sep-2025 12:40 by rmk
; Edited 4-Sep-2025 08:06 by rmk
; Edited 24-May-2025 10:55 by rmk

;; Returns the function that maps a FROMENCODING code to the corresponding MCCS code

```

(CL:WHEN (LISTP FROMENCODING)
  ;; Assume it's a FONTSPEC
  (SETQ FROMENCODING (fetch (FONTSPEC FSFAMILY) of FROMENCODING)))
(if (MEMB FROMENCODING NSFONTFAMILIES)
    then (SETQ FROMENCODING 'XCCS$)
    elseif (MEMB FROMENCODING ALTOFONTFAMILIES)
          then (SETQ FROMENCODING 'ALTOTEXT))
(SELECTQ FROMENCODING

```

```

(XCCS$ (FUNCTION X$TOMCODE))
(ALTOTEXT (FUNCTION ATOMCODE))
(SYMBOL (FUNCTION SYMBOLTOMCODE))
(SIGMA (FUNCTION SIGMATOMCODE))
(MATH (FUNCTION MATHTOMCODE))
(HIPPO (FUNCTION HIPOTOMCODE))
(CYRILLIC (FUNCTION CYRILLICTOMCODE))
(XCCS (FUNCTION XTOMCODE))
(GACHA (FUNCTION GACHATOMCODE))
(PALATINO (FUNCTION PALATINOTOMCODE))
(MCCS NIL)
NIL)

```

(MCCSMAPPAIRS

[LAMBDA (FROMENCODING NOIDENTITY)

```

; Edited 17-Apr-2026 00:02 by rmk
; Edited 14-Apr-2026 00:35 by rmk
; Edited 11-Apr-2026 23:44 by rmk
; Edited 8-Mar-2026 23:43 by rmk
; Edited 3-Mar-2026 23:19 by rmk
; Edited 26-Feb-2026 12:56 by rmk
; Edited 7-Oct-2025 14:47 by rmk
; Edited 6-Oct-2025 09:47 by rmk
; Edited 20-Sep-2025 09:45 by rmk
; Edited 6-Sep-2025 16:43 by rmk
; Edited 31-Aug-2025 16:16 by rmk

```

```

;; Returns the pairs for MOVEFONTCHARS to use to move charset-0 glyphs into their MCCS positions. For example, the Leftarrow and Lowline
;; glyphs switch positions in an XCCS$ font. Returns NIL (= nothing to do) if there is no function.

```

(LET (PAIRS KEEPSCO)

[SETQ PAIRS (SELECTQ FROMENCODING

```

(GACHA [XCCSUNDEFINEDPAIRS '((Lowline ↑X])
(ALTOTEXT ; Wipe the metas
(XCCSUNDEFINEDPAIRS ALTOTEXTTOMCCS))

```

```

(TITAN (XCCSUNDEFINEDPAIRS TITANTOMCCS))
(TITANLEGAL

```

```

(XCCS$ ; Leave the Metas
[XCCS.CS0.UNDEFINED '((Circumflex Uparrow)
(Uparrow Circumflex)
(Lowline Leftarrow)
(Leftarrow Lowline))

```

```

(UNICODE *MCCSTOUNICODE*)
(PALATINO (XCCSUNDEFINEDPAIRS PALATINOTOMCCS))
(PROGN (if (MEMB FROMENCODING '(HIPPO CYRILLIC SYMBOL SIGMA MATH))

```

```

then (SETQ NOIDENTITY T)
(SETQ KEEPSCO NIL)
else (SETQ KEEPSCO T)
(for C M FN from 0 to \MAXTHINCHAR first (CL:UNLESS (SETQ FN (MCCSMAPFN
FROMENCODING
))

```

(RETURN))

when (SETQ M (APPLY* FN C NOIDENTITY)) collect (LIST M C]

(CL:WHEN (LISTP PAIRS)

;; Weed out interspersed comments, convert to charcodes

(CL:UNLESS NOIDENTITY

```

(for DEST from 0 to \MAXTHINCHAR unless (ASSOC DEST PAIRS) do (push PAIRS (LIST DEST DEST
)))

```

[SETQ PAIRS (for P SOURCE in PAIRS when (LISTP P) unless (EQ '* (CAR P))

```

collect (SETQ SOURCE (CADR P))
(LIST (OR (CHARCODEP (CAR P))
(CHARCODE.DECODE (CAR P)))

```

(if (LISTP SOURCE)

then ; Allows for the (Uparrow TERMINAL) case above, for MOVEFONTCHARS

```

(CONS (CL:IF (CHARCODEP (CAR SOURCE))
(CAR SOURCE)
(CHARCODE.DECODE (CAR SOURCE)))

```

(CDR SOURCE)
elseif (CHARCODEP SOURCE)
else (CHARCODE.DECODE SOURCE]

:: If a source is moved to a dest, the default is that that source gets replaced by a slug and there is no separate replacement for that
:: source. That slug may then be coerced from another font. That is, we don't expect two codes to have the same glyph, by default.
:: But families like SYMBOL, HIPPO etc. want to preserve CS0 even if they copy their glyphs also to somewhere else--KEEPCS0.

(CL:UNLESS KEEPCS0

(SETQ PAIRS (APPEND (for C from 0 to \MAXTHINCHAR when [thereis P SCORE in PAIRS
eachtime [SETQ SCORE
(CAR (MKLIST (CADR P])
suchthat

:: C's glyph is moving somewhere else, and nothing
:: is replacing it. Slug it out.

(AND (EQ SCORE C)
(NEQ SCORE (CAR P))
(NOT (ASSOC C PAIRS]

collect (LIST C (CONS NIL)))
PAIRS))
(SETQ PAIRS (SORT PAIRS T)))
[AND NIL (SETQ PAIRS (APPEND PAIRS (for P DEST SOURCE SCORE in PAIRS eachtime (SETQ DEST
(CAR P))
(SETQ SOURCE
(CADR P))
(SETQ SCORE
(CAR (MKLIST
SOURCE
))))

; Already a slug?

when SCORE unless (OR (AND KEEPCS0 (ILEQ SCORE
\MAXTHINCHAR
))
(AND (LISTP SOURCE)
(LITATOM (CADR SOURCE)))
(ASSOC SCORE PAIRS))

collect

:: Don't slugify a source code in this font if it is coming from another font, or if
:: that source is also a destination here.

(LIST SCORE (CONS NIL])

PAIRS])

(XCCS.CS0.UNDEFINED
[LAMBDA (ADDITIONS)

; Edited 8-Mar-2026 23:32 by rmk
; Edited 5-Oct-2025 22:44 by rmk

:: Maps slugs to all undefined/reserved characters in XCCS, then throw in the additions

[SETQ ADDITIONS (for P in ADDITIONS when (LISTP P) unless (EQ '** (CAR P)) collect (LIST (OR (CHARCODEP (CAR P))
(CHARCODE.DECODE
(CAR P)))
(OR (CHARCODEP (CADR P))
(CHARCODE.DECODE
(CADR P])

(APPEND ADDITIONS (for I from 0 to (SUB1 (CHARCODE SPACE)) collect (LIST I NIL))
(for I from (CHARCODE "0,#NULL") to (SUB1 (CHARCODE "0,#SPACE")) collect (LIST I NIL))
(for I in (CHARCODE ("0,177" "0,246" "0,250" "0,300" "0,351" "0,326" "0,327" "0,330" "0,331" "0,332"
"0,333" "0,377"))
collect (LIST I NIL])

(XCCSUNDEFINEDPAIRS
[LAMBDA (ADDITIONS)

; Edited 8-Mar-2026 23:42 by rmk
; Edited 6-Mar-2026 23:57 by rmk
; Edited 5-Oct-2025 22:39 by rmk
; Edited 2-Sep-2025 13:14 by rmk

:: This clears out everything above Ascii in charset 0, allowing any ADDITIONS. XCCS.CS0.UNDEFINED clears out only the truly undefined

```
(LET ((PAIRS (XCCS.CS0.UNDEFINED ADDITIONS)))
  (for I from 128 to \MAXTHINCHAR unless (ASSOC I PAIRS) do (push PAIRS (LIST I NIL)))
  PAIRS])
```

)

(RPAQQ ALTOTEXTTOMCCS

```
:: From bravo doc
("356,055" ↑N MINUS)
("357,44" ↑V ENDASH)
(EMDASH ↑S)
(EMQUAD ↑O)
("356,055" ↑X MINUS)
(FIGURESPLACE ↑Y ENQUAD)
:: Fom current Helvetica/Timesroman fonts
("0,317" "0,1" HACHEK)
("361,255" "0,3" DIARESIS)
("0,310" "0,4" CCEDILLA)
("0,301" "0,5" GRAVE)
("360,41" "0,6" ff)
("0,271" "0,7" LSQ)
("0,241" "0,10" SPANISHEXCL)
("0,302" "0,13" ACUTE)
("0,304" "0,20" TILDE)
("360,42" "0,21" ffi)
("360,43" "0,22" ffi)
("360,44" "0,24" fi)
("360,45" "0,25" fl)
("357,44" "0,26" ENDASH)
("0,306" "0,27" BREVE)
(ENQUAD "0,34")
("0,304" "0,36" TILDE)
("0,251" "0,140")
("361,47" "0,200" A-umlaut)
("361,124" "0,201" O-umlaut)
("361,47" "0,202" A-ring)
("357,44" "0,233" ENDASH)
(EMDASH "0,234")
("361,247" "0,240" a-umlaut)
("361,324" "0,241" o-umlaut)
("361,250" "0,242" a-ring)
("361,345" "0,243" u-umlaut)
(Circumflex "0,254")
("0,242" "0,260" CENTS)
("0,243" "0,261" POUND)
("41,172" "0,265" STAR)
("0,247" "0,266" SECTION)
("357,146" "0,267" BULLET)
("357,60" "0,270" DAGGER)
("357,061" "0,271" DOUBLEDAGGER)
("0,266" "0,272" PARAGRAPH)
("0,261" "0,274" PLUSMINUS)
("0,241" "0,275" SPANISHEXCL)
("0,277" "0,276" SPANISHQUES)
(Lowline "0,277"))
```

(RPAQQ SYMBOLTOMCCS

```
("42,46" "0,162")
("41,145" "0,26")
("41,146" "0,27")
("42,120" "0,55")
("42,121" "0,56")
("0,74" "0,36")
("0,76" "0,37")
(Null "0,1")
("0,264" "0,2")
("41,142" "0,3")
(Null "0,4")
("41,176" "0,5")
("0,261" "0,6")
("357,175" Bell)
("357,142" Backspace)
("357,143" Tab)
("357,144" Linefeed)
("357,145" "0,13")
(Null Page)
("0,270" Newline)
(Null "0,16")
(Null "0,17")
("357,160" "0,20")
("357,162" "0,21")
```

("357,131" "0,22")
("357,130" "0,23")
("41,145" "0,24")
("41,146" "0,25")
(Null "0,26")
(Null "0,27")
("356,176" "0,30")
("357,171" "0,31")
("357,133" "0,32")
("357,132" Escape)
("41,142" "0,34")
("357,163" "0,35")
(Null "0,36")
(Null Tenexeol)
(Null Space)
("0,256" "0,41")
(Circumflex "0,42")
("0,257" "0,43")
("357,122" Dollar)
("357,102" "0,45")
("357,103" "0,46")
("357,167" "0,47")
("357,115" "0,50")
("357,117" "0,51")
(Null "0,52")
(Null "0,53")
("357,116" "0,54")
(Null "0,55")
(Null "0,56")
(Null "0,57")
(Null Zero)
(INFINITY One)
("357,112" Two)
("357,113" Three)
("357,141" Four)
(Null Five)
("357,154" Six)
(Lowline Seven)
("357,265" Eight)
("357,264" Nine)
("357,152" "0,72")
("357,247" "0,73")
(Null "0,74")
(Null "0,75")
(Null "0,76")
("0,57" "0,77")
(Null "0,100")
("357,127" "0,133")
("357,126" "0,134")
(Null "0,135")
("357,266" Uparrow)
("357,267" Leftarrow)
("357,66" "0,140")
("357,67" "0,141")
("357,262" "0,142")
("357,263" "0,143")
("357,260" "0,144")
("357,261" "0,145")
("0,173" "0,146")
("0,175" "0,147")
("357,62" "0,150")
("357,63" "0,151")
("356,174" "0,152")
("41,102" "0,153")
("357,73" "0,154")
("357,72" "0,155")
("42,44" "0,156")
("42,46" "0,157")
("357,174" "0,160")
("41,142" "0,161")
(Null "0,162")
("357,165" "0,163")
(Null "0,164")
(Null "0,165")
(Null "0,166")
(Null "0,167")
("0,247" "0,170")
("357,60" "0,171")
("357,61" "0,172")
("0,266" "0,173")
("0,100" "0,174")
("0,323" "0,175")
("0,243" "0,176")
(Dollar Rubout)
(Null "0,200")
(Null "0,201")

(Null "0,202")
(Null "0,203")
(Null "0,204")
(Null "0,205")
(Null "0,206")
(Null "0,207")
(Null "0,210")
(Null "0,211")
(Null "0,212")
(Null "0,213")
(Null "0,214")
(Null "0,215")
(Null "0,216")
(Null "0,217")
(Null "0,220")
(Null "0,221")
(Null "0,222")
(Null "0,223")
(Null "0,224")
(Null "0,225")
(Null "0,226")
(Null "0,227")
(Null "0,230")
(Null "0,231")
(Null "0,232")
(Null "0,233")
(Null "0,234")
(Null "0,235")
(Null "0,236")
(Null "0,237")
(Null "0,240")
(Null "0,241")
(Null "0,242")
(Null "0,243")
(Null Currency)
(Null "0,245")
(Null "0,246")
(Null "0,247")
(Null "0,250")
(Null "0,251")
(Null LEFT-DOUBLEQUOTE)
(Null "0,253")
(Null Lowline)
(Null Circumflex)
(Null "0,256")
(Null "0,257")
(Null "0,260")
(Null "0,261")
(Null "0,262")
(Null "0,263")
(Null "0,264")
(Null "0,265")
(Null "0,266")
(Null "0,267")
(Null "0,270")
(Null "0,271")
(Null RIGHT-DOUBLEQUOTE)
(Null "0,273")
(Null "0,274")
(Null "0,275")
(Null "0,276")
(Null "0,277")
(Null "0,300")
(Null "0,301")
(Null "0,302")
(Null "0,303")
(Null "0,304")
(Null "0,305")
(Null "0,306")
(Null "0,307")
(Null "0,310")
(Null "0,311")
(Null "0,312")
(Null "0,313")
(Null "0,314")
(Null "0,315")
(Null "0,316")
(Null "0,317")
(Null "0,320")
(Null "0,321")
(Null "0,322")
(Null "0,323")
(Null "0,324")
(Null "0,325")
(Null "0,326")
(Null "0,327")

(Null "0,330")
 (Null "0,331")
 (Null "0,332")
 (Null "0,333")
 (Null "0,334")
 (Null "0,335")
 (Null "0,336")
 (Null "0,337")
 (Null "0,340")
 (Null "0,341")
 (Null "0,342")
 (Null "0,343")
 (Null "0,344")
 (Null "0,345")
 (Null "0,346")
 (Null "0,347")
 (Null "0,350")
 (Null "0,351")
 (Null "0,352")
 (Null "0,353")
 (Null "0,354")
 (Null "0,355")
 (Null "0,356")
 (Null "0,357")
 (Null "0,360")
 (Null "0,361")
 (Null "0,362")
 (Null "0,363")
 (Null "0,364")
 (Null "0,365")
 (Null "0,366")
 (Null "0,367")
 (Null "0,370")
 (Null "0,371")
 (Null "0,372")
 (Null "0,373")
 (Null "0,374")
 (Null "0,375")
 (Null "0,376")
 (Null "0,377"))

(RPAQQ SIGMATOMCCS
 ("0,101" "0,101" low squaredot not in XCCS)
 (Contourintegral "0,103")
 (Intersection "0,111")
 (And "0,114")
 (Summation "0,115")
 (Product "0,120")
 (Radical "0,122")
 (Integral "0,123")
 (Union "0,125")
 (Or "0,126"))

(RPAQQ HIPOTOMCCS
 ("356,55" "0,16")
 (EMQUAD "0,17")
 (EMDASH "0,23")
 ("357,44" "0,26")
 ("356,55" "0,30")
 (ENQUAD "0,31")
 ("Greek,101" "0,101")
 ("Greek,102" "0,102")
 ("Greek,121" "0,103")
 ("Greek,105" "0,104")
 ("Greek,106" "0,105")
 ("Greek,132" "0,106")
 ("Greek,104" "0,107")
 ("Greek,112" "0,110")
 ("Greek,114" "0,111")
 ("Greek,115" "0,113")
 ("Greek,116" "0,114")
 ("Greek,117" "0,115")
 ("Greek,120" "0,116")
 ("Greek,122" "0,117")
 ("Greek,123" "0,120")
 ("Greek,113" "0,121")
 ("Greek,125" "0,122")
 ("Greek,126" "0,123")
 ("Greek,130" "0,124")
 ("Greek,131" "0,125")
 ("Greek,135" "0,127")
 ("Greek,133" "0,130")
 ("Greek,134" "0,131")
 ("Greek,111" "0,132")
 ("Greek,141" "0,141")

("Greek,142" "0,142")
 ("Greek,161" "0,143")
 ("Greek,145" "0,144")
 ("Greek,146" "0,145")
 ("Greek,172" "0,146")
 ("Greek,144" "0,147")
 ("Greek,152" "0,150")
 ("Greek,154" "0,151")
 ("Greek,155" "0,153")
 ("Greek,156" "0,154")
 ("Greek,157" "0,155")
 ("Greek,160" "0,156")
 ("Greek,162" "0,157")
 ("Greek,163" "0,160")
 ("Greek,153" "0,161")
 ("Greek,165" "0,162")
 ("Greek,166" "0,163")
 ("Greek,170" "0,164")
 ("Greek,171" "0,165")
 ("Greek,175" "0,167")
 ("Greek,173" "0,170")
 ("Greek,174" "0,171")
 ("Greek,151" "0,172")
 ("357,44" "0,233")
 (EMDASH "0,234")
 ("357,146" "0,267"))))

(RPAQQ CYRILLICTOMCCS

("Cyrillic,47" Dollar)
 ("Cyrillic,71" "0,52")
 ("41,76" "0,55")
 ("Cyrillic,157" Two)
 ("Cyrillic,127" Four)
 ("Cyrillic,150" Six)
 ("Cyrillic,151" Eight)
 ("0,253" "0,74")
 ("0,273" "0,76")
 ("Cyrillic,77" "0,100")
 ("Cyrillic,41" "0,101")
 ("Cyrillic,42" "0,102")
 ("Cyrillic,76" "0,103")
 ("Cyrillic,45" "0,104")
 ("Cyrillic,46" "0,105")
 ("Cyrillic,66" "0,106")
 ("Cyrillic,44" "0,107")
 ("Cyrillic,101" "0,110")
 ("Cyrillic,52" "0,111")
 ("Cyrillic,53" "0,112")
 ("Cyrillic,54" "0,113")
 ("Cyrillic,55" "0,114")
 ("Cyrillic,56" "0,115")
 ("Cyrillic,57" "0,116")
 ("Cyrillic,60" "0,117")
 ("Cyrillic,61" "0,120")
 ("Cyrillic,67" "0,121")
 ("Cyrillic,62" "0,122")
 ("Cyrillic,63" "0,123")
 ("Cyrillic,64" "0,124")
 ("Cyrillic,65" "0,125")
 ("Cyrillic,43" "0,126")
 ("Cyrillic,50" "0,127")
 ("Cyrillic,75" "0,130")
 ("Cyrillic,100" "0,131")
 ("Cyrillic,51" "0,132")
 ("Cyrillic,152" "0,133")
 ("Cyrillic,0" "0,134")
 ("Cyrillic,153" "0,135")
 ("Cyrillic,74" Uparrow)
 ("Cyrillic,154" Leftarrow)
 ("Cyrillic,0" "0,140")
 ("Cyrillic,121" "0,141")
 ("Cyrillic,122" "0,142")
 ("Cyrillic,176" "0,143")
 ("Cyrillic,125" "0,144")
 ("Cyrillic,126" "0,145")
 ("Cyrillic,146" "0,146")
 ("Cyrillic,124" "0,147")
 ("Cyrillic,161" "0,150")
 ("Cyrillic,132" "0,151")
 ("Cyrillic,133" "0,152")
 ("Cyrillic,134" "0,153")
 ("Cyrillic,135" "0,154")
 ("Cyrillic,136" "0,155")
 ("Cyrillic,137" "0,156")
 ("Cyrillic,140" "0,157")

("Cyrillic,141" "0,160")
 ("Cyrillic,147" "0,161")
 ("Cyrillic,142" "0,162")
 ("Cyrillic,143" "0,163")
 ("Cyrillic,144" "0,164")
 ("Cyrillic,145" "0,165")
 ("Cyrillic,123" "0,166")
 ("Cyrillic,130" "0,167")
 ("Cyrillic,155" "0,170")
 ("Cyrillic,160" "0,171")
 ("Cyrillic,131" "0,172")
 ("Cyrillic,72" "0,173")
 ("Cyrillic,0" "0,174")
 ("Cyrillic,73" "0,175")
 ("Cyrillic,70" "0,176")
 ("Cyrillic,0" Rubout)
 ("Cyrillic,156" "0,217")
 ("357,44" "0,233")
 (EMDASH "0,234")
 ("357,146" "0,267"))

(RPAQQ MATHTOMCCS

((Product "0,1")
 ("357,62" "0,2")
 ("357,63" "0,3")
 (Hairspace "0,4")
 ("0,243" "0,5")
 (Integral "0,6")
 (Contourintegral "0,7")
 ("0,266" "0,13")
 ("357,146" "0,17")
 (Summation "0,23")
 ("357,157" "0,26")
 ("357,60" "0,41")
 ("357,147" "0,42")
 (INFINITY "0,43")
 ("0,242" "0,44")
 ("0,270" "0,45")
 (And "0,46")
 ("357,163" "0,47")
 ("0,302" "0,50")
 (Radical "0,51")
 ("0,307" "0,52")
 ("0,261" "0,53")
 ("357,114" "0,54")
 ("357,175" "0,55")
 ("41,150" "0,56")
 ("357,145" "0,57")
 ("357,147" "0,60")
 ("42,42" "0,61")
 ("42,44" "0,62")
 ("41,176" "0,63")
 ("357,142" "0,64")
 ("357,143" "0,65")
 ("357,144" "0,66")
 ("357,154" "0,67")
 ("41,172" "0,70")
 ("0,307" "0,71")
 ("0,247" "0,72")
 ("356,52" "0,73")
 ("41,145" "0,74")
 ("41,142" "0,75")
 ("41,146" "0,76")
 ("0,277" "0,77")
 ("357,100" "0,100")
 (All "0,101")
 (Member "0,102")
 ("357,254" "0,103")
 ("357,271" "0,104")
 (Exists "0,105")
 ("357,61" "0,106")
 ("357,133" "0,107")
 ("357,137" "0,110")
 ("357,131" "0,111")
 ("357,132" "0,112")
 ("357,136" "0,113")
 ("357,130" "0,114")
 ("360,275" "0,115")
 (Notmember "0,116")
 ("357,141" "0,117")
 ("357,161" "0,120")
 ("357,121" "0,121")
 ("357,256" "0,122")
 ("357,171" "0,123")
 ("357,160" "0,124")

(Union "0,125")
 (Or "0,126")
 ("357,162" "0,127")
 ("0,264" "0,130")
 ("360,272" "0,131")
 ("357,270" "0,132")
 ("41,120" "0,133")
 ("41,121" "0,135")
 ("0,257" "0,136")
 ("0,256" "0,137")
 ("357,247" "0,141")
 ("357,123" "0,142")
 ("0,323" "0,143")
 ("357,272" "0,144")
 ("357,167" "0,145")
 ("357,122" "0,146")
 ("357,117" "0,147")
 ("357,150" "0,150")
 ("357,260" "0,151")
 ("357,261" "0,152")
 ("357,262" "0,153")
 ("357,263" "0,154")
 ("357,110" "0,155")
 ("357,152" "0,156")
 ("357,147" "0,157")
 ("357,66" "0,160")
 ("357,70" "0,161")
 ("0,322" "0,162")
 ("357,76" "0,163")
 ("357,74" "0,164")
 ("357,77" "0,165")
 ("357,75" "0,166")
 ("357,102" "0,167")
 ("357,103" "0,170")
 (Intersection "0,171")
 ("357,67" "0,172")
 ("0,274" "0,173")
 ("0,275" "0,174")
 ("0,276" "0,175")
 ("357,120" "0,176"))))

(RPAQQ PALATINOTOMCCS
 ((Circumflex Uparrow)
 (Uparrow NIL)
 (Lowline Leftarrow)
 (Leftarrow NIL)
 ("361,353" "0,32")
 ("361,260" "0,34")
 ("361,277" "0,35")
 ("361,304" "0,36")
 ("361,153" "0,37")
 ("0,255" "0,136")
 ("0,254" "0,137")
 ("0,240" NIL)
 ("361,047" "0,200")
 ("361,124" "0,201")
 ("361,043" "0,202")
 ("361,077" "0,203")
 ("361,114" "0,204")
 ("361,120" "0,205")
 ("361,121" "0,206")
 ("361,117" "0,207")
 ("361,122" "0,210")
 ("361,134" "0,211")
 ("361,140" "0,212")
 ("361,141" "0,213")
 ("361,145" "0,214")
 ("361,137" "0,215")
 ("361,155" "0,216")
 ("361,160" "0,217")
 ("361,142" "0,220")
 ("361,241" "0,221")
 ("361,243" "0,222")
 ("361,276" "0,223")
 ("361,250" "0,224")
 ("361,320" "0,225")
 ("361,321" "0,226")
 ("361,322" "0,227")
 ("361,322" "0,230")
 ("361,334" "0,231")
 ("361,244" "0,232")
 ("361,341" "0,233")
 ("361,261" "0,234")
 ("361,337" "0,235")
 ("361,262" "0,236"))

("361,255" "0,237")
("361,247" "0,240")
("0,057" "0,244")

; Slash, but should be fraction

("357,243" "0,246")
("0,244" "0,250")
("357,052" "0,254")
("357,053" "0,255")
("360,004" "0,256")
("360,005" "0,257")
(EMDASH "0,261")
("357,060" "0,262")
("357,061" "0,263")
("357,146" "0,267")
("43,262" "0,270")
("357,050" "0,271")
("41,104" "0,274")
("357,101" "0,275")
("357,153" "0,311")
("361,314" "0,314")
("375,261" "0,321")
("361,324" "0,324")
("375,362" "0,325")
("375,363" "0,326")
("0,274" "0,327")
("0,275" "0,330")
("0,264" "0,331")
("0,270" "0,332")
("357,152" "0,333")
("361,265" "0,334")
("0,261" "0,335")
("361,042" "0,336")
("357,044" "0,337")
("361,340" "0,340")
("361,041" "0,344")
("361,345" "0,345")
("361,050" "0,346")
("361,044" "0,347")
("361,355" "0,355")
("361,055" "0,356")
("361,061" "0,357")
("361,360" "0,360")
("361,062" "0,362")
("361,065" "0,364")
("361,060" "0,366")
("361,277" "0,367")
("361,100" "0,375")
("361,104" "0,376"))

(RPAQQ TITANTOMCCS
(("0,242" "0,176" cent)
("0,176" NIL Delete cent)
(Lowline ↑X)
(Lowline "0,277")
("0,55" "0,337")
("0,55" "0,55" Hypehn)
("0,274" "0,74" Quarter)
("0,74" NIL Delete quarter)
("0,275" "0,76" Half)
("0,76" NIL Delete half)))

(RPAQQ TITANLEGALATOMCCS
(("0,247" "0,176" Section)
("0,176" NIL Delete section)
(Lowline ↑X)
(Lowline "0,277")
("0,55" "0,337")
("0,55" "0,55")
("0,260" "0,74" Degree)
("0,74" NIL Delete degree)
("0,266" "0,100" Paragraph)
("0,100" NIL Delete Paragraph)))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS GACHATOMCCSARRAY ALTOTOMCCSARRAY SYMBOLTOMCCSARRAY HIPPTOMCCSARRAY
CYRILLICTOMCCSARRAY MATHTOMCCSARRAY SIGMATOMCCSARRAY PALATINOTOMCCSARRAY TITANTOMCCSARRAY
TITANLEGALATOMCCSARRAY)

)

:: For translation of codes in datastructures (e.g. Tedit)

(RPAQ? GACHATOMCCSARRAY [MCCSCODEMAPARRAY (XCCSUNDEFINEDPAIRS `((Lowline ↑X))

(RPAQ? ALTOTOMCCSARRAY (MCCSCODEMAPARRAY ALTOTEXTTOMCCS))

(RPAQ? SYMBOLTOMCCSARRAY (MCCSCODEMAPARRAY SYMBOLTOMCCS))

(RPAQ? HIPOTOMCCSARRAY (MCCSCODEMAPARRAY HIPOTOMCCS))

(RPAQ? CYRILLICTOMCCSARRAY (MCCSCODEMAPARRAY CYRILLICTOMCCS))

(RPAQ? MATHTOMCCSARRAY (MCCSCODEMAPARRAY MATHTOMCCS))

(RPAQ? SIGMATOMCCSARRAY (MCCSCODEMAPARRAY SIGMATOMCCS))

(RPAQ? PALATINOTOMCCSARRAY (MCCSCODEMAPARRAY PALATINOTOMCCS))

(RPAQ? TITANTOMCCSARRAY (MCCSCODEMAPARRAY (XCCSUNDEFINEDPAIRS TITANTOMCCS)))

(RPAQ? TITANLEGALTOMCCSARRAY (MCCSCODEMAPARRAY (XCCSUNDEFINEDPAIRS TITANLEGALTOMCCS)))

:: Mappings into MCCS: needed for e.g. Tedit coercion. \TEDIT.MCCS.TRANSLATE

(DEFINEQ

(GACHATOMCODE
[LAMBDA (GCODE)

; Edited 7-Mar-2026 23:53 by rmk
 ; Edited 6-Mar-2026 11:19 by rmk
 ; Edited 7-Sep-2025 22:38 by rmk
 ; Edited 3-Sep-2025 23:23 by rmk

:: Gacha did not have a code for circumflex, so there is nothing to map

(OR (CL:WHEN (ILEQ GCODE \MAXTHINCHAR)
(ELT GACHATOMCCSARRAY GCODE))
GCODE])

(SYMBOLTOMCODE
[LAMBDA (SCODE)

; Edited 7-Mar-2026 23:53 by rmk
 ; Edited 7-Sep-2025 22:39 by rmk
 ; Edited 3-Sep-2025 10:21 by rmk
 ; Edited 7-Aug-2025 09:37 by rmk
 ; Edited 1-Jun-2025 07:02 by rmk

(OR (CL:WHEN (ILEQ SCODE \MAXTHINCHAR)
(ELT SYMBOLTOMCCSARRAY SCODE))
SCODE])

(SIGMATOMCODE
[LAMBDA (SCODE)

; Edited 7-Mar-2026 23:54 by rmk
 ; Edited 7-Sep-2025 22:39 by rmk
 ; Edited 3-Sep-2025 10:21 by rmk
 ; Edited 1-Jun-2025 07:02 by rmk
 ; Edited 24-May-2025 10:54 by rmk

(OR (CL:WHEN (ILEQ SCODE \MAXTHINCHAR)
(ELT SIGMATOMCCSARRAY SCODE))
SCODE])

(ATOMCODE
[LAMBDA (ACODE)

; Edited 7-Mar-2026 23:54 by rmk
 ; Edited 7-Sep-2025 22:39 by rmk
 ; Edited 3-Sep-2025 10:21 by rmk
 ; Edited 24-May-2025 09:41 by rmk

(OR (CL:WHEN (ILEQ ACODE \MAXTHINCHAR)
(ELT ALTOTOMCCSARRAY ACODE))
ACODE])

**(MATHTOMCODE
[LAMBDA (MATHCODE)**

; Edited 7-Mar-2026 23:54 by rmk
; Edited 7-Sep-2025 22:39 by rmk
; Edited 4-Sep-2025 08:18 by rmk
; Edited 1-Jun-2025 07:02 by rmk
; Edited 24-May-2025 10:58 by rmk

**(OR (CL:WHEN (ILEQ MATHCODE \MAXTHINCHAR)
(ELT MATHTOMCCSARRAY MATHCODE))
MATHCODE])**

**(HIPOTOMCODE
[LAMBDA (HCODE)**

; Edited 7-Mar-2026 23:54 by rmk
; Edited 7-Sep-2025 22:40 by rmk
; Edited 3-Sep-2025 10:22 by rmk
; Edited 24-May-2025 09:40 by rmk

**(OR (CL:WHEN (ILEQ HCODE \MAXTHINCHAR)
(ELT HIPOTOMCCSARRAY HCODE))
HCODE])**

**(CYRILLICTOMCODE
[LAMBDA (CCODE)**

; Edited 7-Mar-2026 23:55 by rmk
; Edited 7-Sep-2025 22:40 by rmk
; Edited 24-May-2025 09:38 by rmk

**(OR (CL:WHEN (ILEQ CCODE \MAXTHINCHAR)
(ELT CYRILLICTOMCCSARRAY CCODE))
CCODE])**

**(PALATINOTOMCODE
[LAMBDA (PCODE)**

; Edited 7-Mar-2026 23:55 by rmk
; Edited 5-Oct-2025 20:08 by rmk
; Edited 7-Sep-2025 22:39 by rmk
; Edited 3-Sep-2025 10:21 by rmk
; Edited 7-Aug-2025 09:37 by rmk
; Edited 1-Jun-2025 07:02 by rmk

**(OR (CL:WHEN (ILEQ PCODE \MAXTHINCHAR)
(ELT PALATINOTOMCCSARRAY PCODE))
PCODE])**

**(TITANTOMCODE
[LAMBDA (TCODE)**

; Edited 7-Mar-2026 23:51 by rmk
; Edited 6-Mar-2026 11:19 by rmk
; Edited 7-Sep-2025 22:38 by rmk
; Edited 3-Sep-2025 23:23 by rmk
; Edited 30-Aug-2025 21:58 by rmk

**(OR (CL:WHEN (ILEQ TCODE \MAXTHINCHAR)
(ELT TITANTOMCCSARRAY TCODE))
TCODE])**

**(TITANLEGALOMCODE
[LAMBDA (TCODE)**

; Edited 7-Mar-2026 23:52 by rmk
; Edited 6-Mar-2026 11:19 by rmk
; Edited 7-Sep-2025 22:38 by rmk
; Edited 3-Sep-2025 23:23 by rmk
; Edited 30-Aug-2025 21:58 by rmk

**(OR (CL:WHEN (ILEQ TCODE \MAXTHINCHAR)
(ELT TITANLEGALOMCCSARRAY TCODE))
TCODE])**

)

(RPAQ? DISPLAYFONTCOERCIONS
'(((HELVETICA (<= * 2))
(HELVETICA 4))
((MODERN (<= 15 * 16))
(* 14))

```

((MODERN (<= 17 * 21)
 * 18))
((MODERN (<= 22 * 28)
 * 24))
((MODERN (<= 29 * 33)
 * 30))
((MODERN (<= 34 * 40)
 * 36))
((MODERN (<= 41 * 65)
 * 48))
((MODERN (<= 66 *)
 * 72))
((PALATINO 9)
 (PALATINO 12))
((PALATINO (<= * 8))
 (PALATINO 10))
((TITAN (<= * 9)
 BOLD)
 (MODERN 10))
((TITAN (<= * 9)
 ITALIC)
 (MODERN 10))
((TITAN (<= * 9))
 (TITAN 10))
(LPT AMTEX)))

```

(RPAQ? DISPLAYCHARCOERCIONS

```

'((GACHA TERMINAL)
 (MODERN CLASSIC)
 (TIMESROMAN CLASSIC)
 (HELVETICA MODERN)
 (TERMINAL MODERN)
 (HIPPO CLASSIC)
 (CYRILLIC CLASSIC)
 (MATH CLASSIC)
 (SIGMA MODERN)
 (SYMBOL MODERN)
 (TITAN CLASSIC)
 (PALATINO CLASSIC)
 (OPTIMA MODERN)
 (BOLDPS CLASSIC)
 (PCTERMINAL CLASSIC)
 (TITANLEGAL CLASSIC)))

```

:: Defunct coerctions? Mapping for DOS filenames, Adobe equivalences

(RPAQ? ADOBEDISPLAYFONTCOERCIONS

```

'(((HELVETICABLACK 16)
 (HELVETICABLACK 18))
 ((SYMBOL)
 (ADOBESYMBOL))
 ((SYMBOL 11)
 (ADOBESYMBOL 10))
 ((AVANTGARDE-DEMI)
 (AVANTGARDE))
 ((AVANTGARDE-BOOK)
 (AVANTGARDE))
 ((NEWCENTURYSCHLBK)
 (CENTURYSCHOOLBOOK))
 ((BOOKMAN-LIGHT)
 (BOOKMAN))
 ((BOOKMAN-DEMI)
 (BOOKMAN))
 ((HELVETICA-NARROW)
 (HELVETICANARROW))
 ((HELVETICA 24)
 (ADOBEHELVETICA 24))))

```

(RPAQ? *DISPLAY-FONT-NAME-MAP*

```

'((TIMESROMAN . TR)
 (HELVETICA . HV)
 (TIMESROMAND . TD)
 (HELVETICAD . HD)
 (MODERN . MD)
 (CLASSIC . CL)
 (GACHA . GC)
 (TITAN . TI)
 (LETTERGOTHIC . LG)
 (BOLDPS . BP)
 (TERMINAL . TM)
 (CLASSICTHIN . CT)
 (HIPPO . HP)
 (LOGO . LG)
 (MATH . MA)

```

{MEDLEY}<sources>MCCSFONTS.;1 (*DISPLAY-FONT-NAME-MAP* cont.)

Page 17

(OLDENGLISH . OE)
(SYMBOL . SY))

FUNCTION INDEX

ATOMCODE	14	MCCSCODEMAPARRAY	2	TITANLEGALATOMCODE	15
CYRILLICTOMCODE	15	MCCSMAPFN	3	TITANTOMCODE	15
DISPLAYENCODINGFN	2	MCCSMAPPAIRS	4	XCCS.CS0.UNDEFINED	5
GACHATOMCODE	14	PALATINOTOMCODE	15	XCCSUNDEFINEDPAIRS	5
HIPPOTOMCODE	15	SIGMATOMCODE	14		
MATHTOMCODE	15	SYMBOLTOMCODE	14		

VARIABLE INDEX

DISPLAY-FONT-NAME-MAP ..	16	GACHATOMCCSARRAY	14	SIGMATOMCCS	9
ADOBEDISPLAYFONTCOERCIONS	16	HIPPOTOMCCS	9	SIGMATOMCCSARRAY	14
ALTOFONTFAMILIES	2	HIPPOTOMCCSARRAY	14	SYMBOLTOMCCS	6
ALTOTEXTTOMCCS	6	MATHTOMCCS	11	SYMBOLTOMCCSARRAY	14
ALTOTOMCCSARRAY	14	MATHTOMCCSARRAY	14	TITANLEGALATOMCCS	13
CYRILLICTOMCCS	10	MCCSFONTFAMILIES	2	TITANLEGALATOMCCSARRAY	14
CYRILLICTOMCCSARRAY	14	NSFONTFAMILIES	2	TITANTOMCCS	13
DISPLAYCHARCOERCIONS	16	PALATINOTOMCCS	12	TITANTOMCCSARRAY	14
DISPLAYFONTCOERCIONS	15	PALATINOTOMCCSARRAY	14		
