File created: 7-Oct-2025 14:52:10 **{WMEDLEY}<sources>MCCS.;152**

edit by: rmk

changes to: (FNS MCCSMAPPAIRS)

previous date: 6-Oct-2025 16:44:20 {WMEDLEY}<sources>MCCS.;149

Read Table: INTERLISP

Package: INTERLISP

Format: MCCS

```
(RPAQQ MCCSCOMS
       [;; Stringlet number encoding common to MCCS and XCCS
        (FNS \MCCSINCCODE \MCCSPEEKCCODE \MCCSOUTCHAR \MCCSBACKCCODE \MCCSFORMATBYTESTREAM
             \MCCSCHARSETFN)
        (FNS \CREATE.MCCS.EXTERNALFORMAT)
        (FNS \MCCS.24BITENCODING.ERROR)
        (INITVARS (*SIGNAL-MCCS.24BITENCODING.ERROR*))
        (DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (CONSTANTS (\NORUNCODE 255)
                                                            (NSCHARSETSHIFT 255))
                                                 (MACROS \RUNCODED)))
        (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\CREATE.MCCS.EXTERNALFORMAT :MCCS)
                                           (\CREATE.MCCS.EXTERNALFORMAT :XCCS)))
        ;;
        ;; Assignment of MCCS characters
        (ALISTS (CHARACTERNAMES Lowline Circumflex Currency Leftarrow Uparrow Dollar Underline))
        ;; Mapping between true XCCS and MCCS codes
        (FNS MTOXCODE XTOMCODE XTOMSTRING MTOXSTRING)
        (FNS MTOX$CODE X$TOMCODE)
        (FNS KANJICHARSETP CHINESECHARSETP)
        (COMS                                                   ;  Mapping to MCCS
              (VARS ALTOTEXT2MCCS SYMBOLTOMCCS SIGMATOMCCS HIPPOTOMCCS CYRILLICTOMCCS MATHTOMCCS
                    PALATINOTOMCCS)
              (FNS MCCSCODEMAPARRAY)
              (GLOBALVARS ALTOTOMCCSARRAY SYMBOLTOMCCSARRAY HIPPOTOMCCSARRAY CYRILLICTOMCCSARRAY
                    MATHTOMCCSARRAY SIGMATOMCCSARRAY PALATINOTOMCCSARRAY)
              (INITVARS (ALTOTOMCCSARRAY (MCCSCODEMAPARRAY 'MCCS))
                    (SYMBOLTOMCCSARRAY (MCCSCODEMAPARRAY SYMBOLTOMCCS))
                    (HIPPOTOMCCSARRAY (MCCSCODEMAPARRAY HIPPOTOMCCS))
                    (CYRILLICTOMCCSARRAY (MCCSCODEMAPARRAY CYRILLICTOMCCS))
                    (MATHTOMCCSARRAY (MCCSCODEMAPARRAY MATHTOMCCS))
                    (SIGMATOMCCSARRAY (MCCSCODEMAPARRAY SIGMATOMCCS))
                    (PALATINOTOMCCSARRAY (MCCSCODEMAPARRAY PALATINOTOMCCS)))
              (FNS MCCSMAPFN MCCSMAPPAIRS XCCS.CS0.UNDEFINED XCCSUNDEFINEDPAIRS)
              (COMS ;; Mappings into MCCS: needed for hardcopy and Tedit coercion, e.g. \TEDIT.MCCS.TRANSLATE
                    (FNS GACHATOMCODE SYMBOLTOMCODE SIGMATOMCODE ATOMCODE MATHTOMCODE HIPPOTOMCODE
                         CYRILLICTOMCODE PALATINOTOMCODE])

;; Stringlet number encoding common to MCCS and XCCS

(DEFINEQ

(\MCCSINCCODE
  [LAMBDA (STREAM COUNTP)                                       ; Edited  9-Sep-2025 22:42 by rmk
                                                                ; Edited  8-Dec-2023 15:28 by rmk
                                                                ; Edited  6-Aug-2021 15:57 by rmk:

;;; Returns a 16 bit character code.  SHIFTEDCSET is STREAM's char set left shifted 8.

;;; If COUNTP is non-NIL, the variable *BYTECOUNTER* is set freely to the number of bytes read.

;;; This doesn't do  EOL conversion, \INCHAR does that

    (DECLARE (USEDFREE *BYTECOUNTER*))
    (\DTEST STREAM 'STREAM)
    (LET (NUMBYTES (CSET (ffetch (STREAM CHARSET) of STREAM))
                (CHAR (\BIN STREAM)))                           ; Error on EOF unless ENDOFSTREAMOP does something
                                                                ; else.

        ;;  NUMBYTES tracks the number of \BINs.

        (IF (EQ CHAR NSCHARSETSHIFT)
            THEN                                                ; Shifting character sets, toss CHAR
                 (SETQ CSET (\BIN STREAM))
                 (IF (NEQ NSCHARSETSHIFT CSET)
                     THEN                                       ; Shift to new runcode CSET: SH CS CH
```

```
                    (SETQ CHAR (\BIN STREAM))
                    (SETQ NUMBYTES 3)
                    (freplace (STREAM CHARSET) of STREAM with CSET)
                ELSEIF (EQ 0 (\BIN STREAM))
                  THEN                                          ; SH SH CSH  CS CH where CSH is 0

                    ;; The high-order character set byte must be 0, because we don't support obese characters (24 bit)

                    (SETQ CSET (\BIN STREAM))
                    (SETQ CHAR (\BIN STREAM))                   ; To align with below
                    (SETQ NUMBYTES 5)
                    (freplace (STREAM CHARSET) of STREAM with \NORUNCODE)
                ELSE (\MCCS.24BITENCODING.ERROR STREAM))

              ;; The stream now knows the new character set, runcoded or not.


          ELSEIF (EQ CSET \NORUNCODE)
            THEN                                                ; 2-bytes
                (SETQ CSET CHAR)
                (SETQ CHAR (\BIN STREAM))
                (SETQ NUMBYTES 2)
          ELSE ;; Runcoded CSET and CHAR
               (SETQ NUMBYTES 1))
        (CL:WHEN COUNTP (SETQ *BYTECOUNTER* NUMBYTES))
        (CL:WHEN CHAR                                           ; Typically NIL if ENDOFSTREAMOP returned NIL at EOF
             (LOGOR (UNFOLD CSET 256)
                    CHAR))])


(\MCCSPEEKCCODE
  [LAMBDA (STREAM NOERROR)                                      ; Edited  9-Sep-2025 22:43 by rmk
                                                                ; Edited 23-Apr-2025 14:16 by rmk
                                                                ; Edited  8-Dec-2023 15:32 by rmk
                                                                ; Edited 21-Jun-2021 23:44 by rmk:

    ;; Modeled on \MCCSINCCODE, but peeks at the last byte in the sequence, leaves the stream unchanged

    (\DTEST STREAM 'STREAM)
    (LET ((CSET (ffetch (STREAM CHARSET) of STREAM))
          (CHAR (\PEEKBIN STREAM NOERROR)))

        ;; Returns a 16 bit character code.  Doesn't do EOL conversion--\PEEKCCODE does that.

        ;; We don't change the charset in the stream, put the file ptr back the way it was.

        (CL:WHEN CHAR
             (IF (EQ CHAR NSCHARSETSHIFT)
                 THEN (\BIN STREAM)                             ; Read the peeked shifting byte
                      (SETQ CSET (\BIN STREAM))                 ; Consume the char shift byte
                      (IF (NEQ CSET NSCHARSETSHIFT)
                          THEN ;; Shift to new runcode CSET: SH CS CH.  We have to BIN what we peeked, BIN, and peek again
                               (SETQ CHAR (\PEEKBIN STREAM NOERROR))
                               (\BACKFILEPTR STREAM)
                               (\BACKFILEPTR STREAM)
                        ELSEIF (EQ 0 (\BIN STREAM))
                          THEN                                  ; SH SH CSH  CS CH where CSH is 0

                               ;; Note: no eof error check on this \BIN -- an eof in the middle of a charset shift is an error

                               (SETQ CSET (\BIN STREAM))
                               (SETQ CHAR (\PEEKBIN STREAM NOERROR))
                               (\BACKFILEPTR STREAM)
                               (\BACKFILEPTR STREAM)
                               (\BACKFILEPTR STREAM)
                               (\BACKFILEPTR STREAM)
                        ELSE (\MCCS.24BITENCODING.ERROR STREAM))
                  ELSEIF (EQ CSET \NORUNCODE)
                    THEN                                        ; 2 byte runs, BIN/PEEK/BACK
                         (SETQ CSET CHAR)
                         (\BIN STREAM)
                         (SETQ CHAR (\PEEKBIN STREAM NOERROR))
                                                                ; One BACKFILEPTR seems OK
```

```
                    (\BACKFILEPTR STREAM))

        ;; No need to back up for the runcoded case

        (CL:WHEN CHAR
              (LOGOR (UNFOLD CSET 256)
                     CHAR)))])


(\MCCSOUTCHAR
  [LAMBDA (STREAM CHARCODE)                          ; Edited 23-Apr-2025 14:16 by rmk
                                                     ; Edited 13-Aug-2021 10:24 by rmk:

    ;; Closed function for the :MCCS external format

    (COND
       ((EQ CHARCODE (CHARCODE EOL))
        (FREPLACE (STREAM CHARPOSITION) OF STREAM WITH 0)
        [COND
           [(NOT (\RUNCODED STREAM))                 ; Charset is a constant 0, we put out the high-order byte.
            (\BOUT STREAM (\CHARSET (CHARCODE EOL]
           ((EQ (\CHARSET (CHARCODE EOL))
                (ffetch (STREAM CHARSET) of STREAM)))
           (T                                        ; We are runcoded, and not in character set 0, have to shift.
              (\BOUT STREAM NSCHARSETSHIFT)
              (\BOUT STREAM (freplace (STREAM CHARSET) of STREAM with (\CHARSET (CHARCODE EOL]

        ;; We are now in the right charset (0) for the first EOL byte.  For CRLF, the CR is immediately followed by the LF byte, without the prefix 0 byte
        ;; even if not runcoded, i.e. the 2 bytes are though of as a composite.  The stream is left in CSET0 (the freplace above), read for another shift
        ;; according to the next shift in a runcoded file.

        (\BOUTEOL STREAM))
       (T (CHANGE (FFETCH (STREAM CHARPOSITION) OF STREAM)
                  (IPLUS16 1 DATUM))
          (COND
             ((NOT (\RUNCODED STREAM))
              (\BOUT STREAM (\CHARSET CHARCODE))
              (\BOUT STREAM (\CHAR8CODE CHARCODE)))
             ((EQ (\CHARSET CHARCODE)
                  (ffetch (STREAM CHARSET) of STREAM))
              (\BOUT STREAM (\CHAR8CODE CHARCODE)))
             (T (\BOUT STREAM NSCHARSETSHIFT)
                (\BOUT STREAM (freplace (STREAM CHARSET) of STREAM with (\CHARSET CHARCODE)))
                (\BOUT STREAM (\CHAR8CODE CHARCODE])


(\MCCSBACKCCODE
  [LAMBDA (STREAM COUNTP)                            ; Edited  8-Dec-2023 15:34 by rmk
                                                     ; Edited 19-Jul-2022 17:12 by rmk
                                                     ; Edited 13-Aug-2021 14:08 by rmk:
    (DECLARE (USEDFREE *BYTECOUNTER*))
    (LET ((BYTE (AND (\BACKFILEPTR STREAM)
                     (\PEEKBIN STREAM)))
          (CSET (fetch (STREAM CHARSET) of STREAM)))
         (CL:WHEN BYTE

             ;; The immediately preceding byte must be a character byte. If it is a byte in a runcode, then we are done, even if the byte before is part
             ;; of a shift sequence.

             ;; But if we are currently in a nonruncoded file, we have to go back one more to get the character set byte.

             ;; If we can't back up, we are already at the beginning.

             (IF (EQ \NORUNCODE CSET)
                 THEN (IF (\BACKFILEPTR STREAM)
                          THEN (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -2))
                               (LOGOR (UNFOLD (\PEEKBIN STREAM)
                                              256)
                                      BYTE)
                          ELSE (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -1))
                               NIL)
                 ELSE (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -1))
                      (LOGOR (UNFOLD CSET 256)
                             BYTE)))])
```

```
(\MCCSFORMATBYTESTREAM
  [LAMBDA (STREAM BYTESTREAM)                    ; Edited 27-May-2025 23:42 by rmk
                                                 ; Edited 26-Mar-2024 11:00 by rmk
                                                 ; Edited 19-Mar-2024 16:02 by rmk
    (\EXTERNALFORMAT BYTESTREAM (\EXTERNALFORMAT STREAM))

    ;; This stream may be read as a continuation of STREAM (TTYIN, LAFITE?), and we want to make sure that the bytes are encoded properly.  So
    ;; let's assert (and possibly mark) that that's its current situation.

    (\MCCSCHARSETFN BYTESTREAM (fetch (STREAM CHARSET) of STREAM))
    BYTESTREAM])


(\MCCSCHARSETFN
  [LAMBDA (STREAM CHARSET DONTMARKSTREAM)        ; Edited  9-Dec-2023 11:18 by rmk

    ;; This differs from  \GENERIC.CHARSET in that it actually writes the shifting bytes into an output stream, unless DONTMARKSTREAM.  It will do
    ;; write the shifts, even if it just replicates the situation that is already there (presumably CHARSET = the old CHARSET).  The client should test and
    ;; avoid calling if useless shifts are not desired.

    (LET [(CSET (ffetch (STREAM CHARSET) of (\DTEST STREAM 'STREAM]
         (CL:WHEN CHARSET
                (CL:WHEN (EQ CHARSET T)
                       (SETQ CHARSET \NORUNCODE))
                (CL:UNLESS (EQ CHARSET CSET)
                       (freplace (STREAM CHARSET) of STREAM with CHARSET)
                       (CL:UNLESS DONTMARKSTREAM
                              (CL:WHEN (\IOMODEP STREAM 'OUTPUT T)
                                     (\BOUT STREAM NSCHARSETSHIFT)
                                     (if (EQ CHARSET \NORUNCODE)
                                           then (\BOUT STREAM \NORUNCODE)
                                                (\BOUT STREAM 0)
                                       else (\BOUT STREAM CHARSET))))))
         CSET])

)

(DEFINEQ

(\CREATE.MCCS.EXTERNALFORMAT
  [LAMBDA (NAME EOL)                             ; Edited 23-Apr-2025 14:19 by rmk
                                                 ; Edited  7-Dec-2023 23:03 by rmk
                                                 ; Edited 30-Jun-2022 18:08 by rmk
                                                 ; Edited 10-Sep-2021 19:49 by rmk:

;;; Create the :MCCS external format.  Stream's EOL overrides the (vacuous) default here

    (MAKE-EXTERNALFORMAT (OR NAME :MCCS)
            (FUNCTION \MCCSINCCODE)
            (FUNCTION \MCCSPEEKCCODE)
            (FUNCTION \MCCSBACKCCODE)
            (FUNCTION \MCCSOUTCHAR)
            (FUNCTION \MCCSFORMATBYTESTREAM)
            (OR EOL 'LF)
            T NIL NIL (FUNCTION \MCCSCHARSETFN])

)

(DEFINEQ

(\MCCS.24BITENCODING.ERROR
  [LAMBDA (STREAM)                               ; Edited  9-Sep-2025 22:41 by rmk
                                                 ; Edited 23-Apr-2025 14:34 by rmk
                                                 (* bvm%: "12-Mar-86 15:35")
    (DECLARE (USEDFREE *SIGNAL-MCCS.24BITENCODING.ERROR*))

;;; Called if we see the sequence shift,shift on STREAM -- means shift to 24-bit character set, which we don't support.  Usually this just means we're
;;; erroneously reading a binary file as text.  If this function returns, its value is taken as a character set to shift to

    (CL:WHEN *SIGNAL-MCCS.24BITENCODING.ERROR*   ; Only cause error if user/reader cares
         (ERROR "24-bit MCCS encoding not supported" STREAM))
                                                 ; Return charset zero
```

0])

)

(RPAQ? ***SIGNAL-MCCS.24BITENCODING.ERROR*** **)**

(DECLARE%: EVAL@COMPILE DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(RPAQQ **NORUNCODE 255)**

(RPAQQ NSCHARSETSHIFT 255)

(CONSTANTS (\NORUNCODE 255)
(NSCHARSETSHIFT 255))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \RUNCODED MACRO (OPENLAMBDA (STREAM)
;; returns NIL is the stream is not runcoded, that is, if the stream has 16 bit bytes explicitly represented
; note that neq is ok since charsets are known to be SMALLP's
(NEQ **(fetch CHARSET of STREAM)**
\NORUNCODE)))
)
)

;; END EXPORTED DEFINITIONS

(DECLARE%: DONTEVAL@LOAD DOCOPY

(**CREATE.MCCS.EXTERNALFORMAT** :MCCS)

(\CREATE.MCCS.EXTERNALFORMAT :XCCS)
)

;;
;; Assignment of MCCS characters

(ADDTOVAR **CHARACTERNAMES (Lowline "0,254")**
(Circumflex "0,255")
(Currency "0,244")
(Leftarrow "0,137")
(Uparrow "0,136")
(Dollar "0,44")
(Underline Lowline))

;; Mapping between true XCCS and MCCS codes

(DEFINEQ

(MTOXCODE
[LAMBDA (MCODE)

; Edited 7-Sep-2025 22:36 by rmk
; Edited 31-Aug-2025 14:24 by rmk
; Edited 1-May-2025 20:05 by rmk
; Edited 27-Apr-2025 13:42 by rmk

;; Inverts XTOMCODE. Presumably for the \OUTCHAR function of hardcopy devices (like Interpress) that want XCCS codes.

(OR **[**CDR **(**ASSOC **MCODE (**CONSTANT **(for X M from 0 to \MAXTHINCHAR when (SETQ M (XTOMCODE X))**
unless (EQ M X) collect (CONS M X]
MCODE])

(XTOMCODE
[LAMBDA (XCODE)

; Edited 7-Sep-2025 22:36 by rmk
; Edited 4-Sep-2025 00:25 by rmk

(OR **[**CDR **(**ASSOC **XCODE (**CONSTANT **(**APPEND **(**CHARCODE **((Currency . Dollar)**
(Dollar . Currency)))

```
                                    (for X M from 0 to \MAXTHINCHAR when (SETQ M (X$TOMCODE X))
                                       unless (EQ X M) collect (CONS X M]
        XCODE])


(XTOMSTRING
  [LAMBDA (XSTRING DESTRUCTIVE)                          ; Edited  2-Sep-2025 12:14 by rmk
                                                         ; Edited 29-Apr-2025 13:08 by rmk

    ;; Converts Unicodes to MCCS codes in XSTRING.

    (for I XCODE (MSTRING ← (CL:IF DESTRUCTIVE
                                XSTRING
                                (CONCAT XSTRING)))
       from 1 while (SETQ XCODE (NTHCHARCODE XSTRING I)) do (RPLCHARCODE MSTRING I (XTOMCODE XCODE))
       finally (RETURN MSTRING])


(MTOXSTRING
  [LAMBDA (MSTRING DESTRUCTIVE)                          ; Edited  2-Sep-2025 12:22 by rmk
                                                         ; Edited 29-Apr-2025 13:08 by rmk

    ;; Converts XCCS to MCCS codes in XSTRING.

    (for I MCODE (XSTRING ← (CL:IF DESTRUCTIVE
                                MSTRING
                                (CONCAT MSTRING)))
       from 1 while (SETQ MCODE (NTHCHARCODE MSTRING I)) do (RPLCHARCODE XSTRING I (MTOXCODE MCODE))
       finally (RETURN XSTRING])

)

(DEFINEQ

(MTOX$CODE
  [LAMBDA (MCODE)                                        ; Edited  7-Sep-2025 22:37 by rmk
                                                         ; Edited 31-Aug-2025 14:23 by rmk
                                                         ; Edited  7-Aug-2025 08:13 by rmk
                                                         ; Edited 11-May-2025 16:54 by rmk

    ;; Inverts X$TOMCODE.  Only worries about charset 0

    (OR [CDR (ASSOC MCODE (CONSTANT (for X M from 0 to \MAXTHINCHAR when (SETQ M (X$TOMCODE X))
                                       unless (EQ M X) collect (CONS X M]
        MCODE])


(X$TOMCODE
  [LAMBDA (X$CODE)                                       ; Edited  7-Sep-2025 22:37 by rmk
                                                         ; Edited  3-Sep-2025 17:26 by rmk
                                                         ; Edited 31-Aug-2025 11:49 by rmk
                                                         ; Edited  7-Aug-2025 08:14 by rmk

    ;; Swaps arrows with lowline and cirumflex           ; Edited 11-May-2025 16:54 by rmk
    (OR [CAR (find PAIR in (CHARCODE ((Uparrow Circumflex)
                                      (Circumflex Uparrow)
                                      (Leftarrow Lowline)
                                      (Lowline Leftarrow)))
             suchthat (EQ X$CODE (CADR PAIR]
        X$CODE])

)

(DEFINEQ

(KANJICHARSETP
  [LAMBDA (CHARSET)                                      ; Edited 13-Jun-2025 16:33 by rmk

    ;; Returns CHARSET if it is a charset with MCCS Kanji characters

    (AND (<= 48 CHARSET 118)
         CHARSET])
```

```
(CHINESECHARSETP
  [LAMBDA (CHARSET)                                    ; Edited 18-Jun-2025 23:09 by rmk
                                                       ; Edited 13-Jun-2025 16:33 by rmk

    ;; Returns CHARSET if it is a charset with MCCS Chinese characters

    (AND (<= 161 CHARSET 212)
         CHARSET])

)


;;  Mapping to MCCS

(RPAQQ ALTOTEXT2MCCS
       (;; From bravo doc
        (↑N "356,055" MINUS)
        (↑V "357,44" ENDASH)
        (↑S EMDASH)
        (↑O EMQUAD)
        (↑X "356,055" MINUS)
        (↑Y FIGURESPACE ENQUAD)
        ;; Fom current Helvetica/Timesroman fonts
        ("0,1" "0,317" HACHEK)
        ("0,3" "361,255" DIARESIS)
        ("0,4" "0,310" CCEDILLA)
        ("0,5" "0,301" GRAVE)
        ("0,6" "360,41" ff)
        ("0,7" "0,271" LSQ)
        ("0,10" "0,241" SPANISHEXCL)
        ("0,13" "0,302" ACUTE)
        ("0,20" "0,304" TILDE)
        ("0,21" "360,42" ffi)
        ("0,22" "360,43" ffl)
        ("0,24" "360,44" fi)
        ("0,25" "360,45" fl)
        ("0,26" "357,44" ENDASH)
        ("0,27" "0,306" BREVE)
        ("0,34" ENQUAD)
        ("0,36" "0,304" TILDE)
        ("0,140" "0,251")
        ("0,200" "361,47" A-umlaut)
        ("0,201" "361,124" O-umlaut)
        ("0,202" "361,47" A-ring)
        ("0,233" "357,44" ENDASH)
        ("0,234" EMDASH)
        ("0,240" "361,247" a-umlaut)
        ("0,241" "361,324" o-umlaut)
        ("0,242" "361,250" a-ring)
        ("0,243" "361,345" u-umlaut)
        ("0,254" Circumflex)
        ("0,260" "0,242" CENTS)
        ("0,261" "0,243" POUND)
        ("0,265" "41,172" STAR)
        ("0,266" "0,247" SECTION)
        ("0,267" "357,146" BULLET)
        ("0,270" "357,60" DAGGER)
        ("0,271" "357,061" DOUBLEDAGGER)
        ("0,272" "0,266" PARAGRAPH)
        ("0,274" "0,261" PLUSMINUS)
        ("0,275" "0,241" SPANISHEXCL)
        ("0,276" "0,277" SPANISHQUES)
        ("0,277" Lowline)))

(RPAQQ SYMBOLTOMCCS
       (("0,1" Null)
        ("0,2" "0,264")
        ("0,3" "41,142")
        ("0,4" Null)
        ("0,5" "41,176")
        ("0,6" "0,261")
        (Bell "357,175")
        (Backspace "357,142")
        (Tab "357,143")
        (Linefeed "357,144")
        ("0,13" "357,145")
        (Page Null)
        (Newline "0,270")
        ("0,16" Null)
        ("0,17" Null)
        ("0,20" "357,160")
        ("0,21" "357,162")
        ("0,22" "357,131")
        ("0,23" "357,130")
```

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

(RPAQQ SIGMATOMCCS
       (("0,101" "0,101" low squaredot not in XCCS)
        ("0,103" "357,166" contourintegral)
        ("0,111" "357,126" intersection)
        ("0,114" "357,266" and)
        ("0,115" "357,172" Summation)
        ("0,120" "357,173" Product)
        ("0,122" "357,174" radical)
        ("0,123" "357,165" integral)
        ("0,125" "357,127" union)
        ("0,126" "357,267" or)))

(RPAQQ HIPPOTOMCCS
       (("0,16" "356,55")
        ("0,17" EMQUAD)
        ("0,23" EMDASH)
        ("0,26" "357,44")
        ("0,30" "356,55")
        ("0,31" ENQUAD)
        ("0,101" "Greek,101")
        ("0,102" "Greek,102")
        ("0,103" "Greek,121")
        ("0,104" "Greek,105")
        ("0,105" "Greek,106")
        ("0,106" "Greek,132")
        ("0,107" "Greek,104")
        ("0,110" "Greek,112")
        ("0,111" "Greek,114")
        ("0,113" "Greek,115")
        ("0,114" "Greek,116")
        ("0,115" "Greek,117")
        ("0,116" "Greek,120")
        ("0,117" "Greek,122")
        ("0,120" "Greek,123")
        ("0,121" "Greek,113")
        ("0,122" "Greek,125")
        ("0,123" "Greek,126")
        ("0,124" "Greek,130")
        ("0,125" "Greek,131")
        ("0,127" "Greek,135")
        ("0,130" "Greek,133")
        ("0,131" "Greek,134")
        ("0,132" "Greek,111")
        (Uparrow Circumflex)
        (Leftarrow Lowline)
        ("0,141" "Greek,141")
        ("0,142" "Greek,142")
        ("0,143" "Greek,161")
        ("0,144" "Greek,145")
```

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

(RPAQQ CYRILLICTOMCCS
           ((Dollar "Cyrillic,47")
            ("0,52" "Cyrillic,71")
            ("0,55" "41,76")
            (Two "Cyrillic,157")
            (Four "Cyrillic,127")
            (Six "Cyrillic,150")
            (Eight "Cyrillic,151")
            ("0,74" "0,253")
            ("0,76" "0,273")
            ("0,100" "Cyrillic,77")
            ("0,101" "Cyrillic,41")
            ("0,102" "Cyrillic,42")
            ("0,103" "Cyrillic,76")
            ("0,104" "Cyrillic,45")
            ("0,105" "Cyrillic,46")
            ("0,106" "Cyrillic,66")
            ("0,107" "Cyrillic,44")
            ("0,110" "Cyrillic,101")
            ("0,111" "Cyrillic,52")
            ("0,112" "Cyrillic,53")
            ("0,113" "Cyrillic,54")
            ("0,114" "Cyrillic,55")
            ("0,115" "Cyrillic,56")
            ("0,116" "Cyrillic,57")
            ("0,117" "Cyrillic,60")
            ("0,120" "Cyrillic,61")
            ("0,121" "Cyrillic,67")
            ("0,122" "Cyrillic,62")
            ("0,123" "Cyrillic,63")
            ("0,124" "Cyrillic,64")
            ("0,125" "Cyrillic,65")
            ("0,126" "Cyrillic,43")
            ("0,127" "Cyrillic,50")
            ("0,130" "Cyrillic,75")
            ("0,131" "Cyrillic,100")
            ("0,132" "Cyrillic,51")
            ("0,133" "Cyrillic,152")
            ("0,134" "Cyrillic,0")
            ("0,135" "Cyrillic,153")
            (Uparrow "Cyrillic,74")
            (Leftarrow "Cyrillic,154")
            ("0,140" "Cyrillic,0")
            ("0,141" "Cyrillic,121")
            ("0,142" "Cyrillic,122")
            ("0,143" "Cyrillic,176")
            ("0,144" "Cyrillic,125")
            ("0,145" "Cyrillic,126")
            ("0,146" "Cyrillic,146")
            ("0,147" "Cyrillic,124")
            ("0,150" "Cyrillic,161")
            ("0,151" "Cyrillic,132")
            ("0,152" "Cyrillic,133")
            ("0,153" "Cyrillic,134")
            ("0,154" "Cyrillic,135")
            ("0,155" "Cyrillic,136")
            ("0,156" "Cyrillic,137")
            ("0,157" "Cyrillic,140")
            ("0,160" "Cyrillic,141")
            ("0,161" "Cyrillic,147")
            ("0,162" "Cyrillic,142")
            ("0,163" "Cyrillic,143")
            ("0,164" "Cyrillic,144")
```

```
            ("0,165" "Cyrillic,145")
            ("0,166" "Cyrillic,123")
            ("0,167" "Cyrillic,130")
            ("0,170" "Cyrillic,155")
            ("0,171" "Cyrillic,160")
            ("0,172" "Cyrillic,131")
            ("0,173" "Cyrillic,72")
            ("0,174" "Cyrillic,0")
            ("0,175" "Cyrillic,73")
            ("0,176" "Cyrillic,70")
            (Rubout "Cyrillic,0")
            ("0,217" "Cyrillic,156")
            ("0,233" "357,44")
            ("0,234" EMDASH)
            ("0,267" "357,146")))

(RPAQQ MATHTOMCCS
          (("0,1" "357,173")
           ("0,2" "357,62")
           ("0,3" "357,63")
           ("0,4" Null)
           ("0,5" "0,243")
           ("0,6" "357,165")
           (Bell "357,166")
           (Backspace Null)
           (Tab Null)
           (Linefeed Null)
           ("0,13" "0,266")
           (Page Null)
           (Newline Null)
           ("0,16" Null)
           ("0,17" "357,146")
           ("0,20" Null)
           ("0,21" Null)
           ("0,22" Null)
           ("0,23" "357,172")
           ("0,24" Null)
           ("0,25" Null)
           ("0,26" "357,157")
           ("0,27" Null)
           ("0,30" Null)
           ("0,31" Null)
           ("0,32" Null)
           (Escape Null)
           ("0,34" Null)
           ("0,35" Null)
           ("0,36" Null)
           (Tenexeol Null)
           ("0,41" "357,60")
           ("0,42" "357,147")
           ("0,43" INFINITY)
           (Dollar "0,242")
           ("0,45" "0,270")
           ("0,46" "357,266")
           ("0,47" "357,163")
           ("0,50" "0,302")
           ("0,51" "357,174")
           ("0,52" "0,307")
           ("0,53" "0,261")
           ("0,54" "357,114")
           ("0,55" "357,175")
           ("0,56" "41,150")
           ("0,57" "357,145")
           (Zero "357,147")
           (One "42,42")
           (Two "42,44")
           (Three "41,176")
           (Four "357,142")
           (Five "357,143")
           (Six "357,144")
           (Seven "357,154")
           (Eight "41,172")
           (Nine "0,307")
           ("0,72" "0,247")
           ("0,73" Null)
           ("0,74" "41,145")
           ("0,75" "41,142")
           ("0,76" "41,146")
           ("0,77" "0,277")
           ("0,100" "357,100")
           ("0,101" "357,265")
           ("0,102" "357,112")
           ("0,103" "357,254")
           ("0,104" "357,271")
           ("0,105" "357,264")
           ("0,106" "357,61")
           ("0,107" "357,133")
```

```
("0,110" "357,137")
("0,111" "357,131")
("0,112" "357,132")
("0,113" "357,136")
("0,114" "357,130")
("0,115" "360,275")
("0,116" "357,113")
("0,117" "357,141")
("0,120" "357,161")
("0,121" "357,121")
("0,122" "357,256")
("0,123" "357,171")
("0,124" "357,160")
("0,125" "357,127")
("0,126" "357,267")
("0,127" "357,162")
("0,130" "0,264")
("0,131" "360,272")
("0,132" "357,270")
("0,133" Null)
("0,134" Null)
("0,135" Null)
(Uparrow "0,257")
(Leftarrow "0,256")
("0,140" Null)
("0,141" "357,247")
("0,142" "357,123")
("0,143" "0,323")
("0,144" "357,272")
("0,145" "357,167")
("0,146" "357,122")
("0,147" "357,117")
("0,150" "357,150")
("0,151" "357,260")
("0,152" "357,261")
("0,153" "357,262")
("0,154" "357,263")
("0,155" "357,110")
("0,156" "357,152")
("0,157" "357,147")
("0,160" "357,66")
("0,161" "357,70")
("0,162" "0,322")
("0,163" "357,76")
("0,164" "357,74")
("0,165" "357,77")
("0,166" "357,75")
("0,167" "357,102")
("0,170" "357,103")
("0,171" "357,126")
("0,172" "357,67")
("0,173" "0,274")
("0,174" "0,275")
("0,175" "0,276")
("0,176" "357,120")
(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" Null)))

(RPAQQ PALATINOTOMCCS
       ((("0,32" "361,353")
        ("0,34" "361,260")
        ("0,35" "361,277")
        ("0,36" "361,304")
        ("0,37" "361,153")
        ("0,136" "0,255")
        ("0,137" "0,254")
        (NIL "0,240")
        ("0,200" "361,047")
        ("0,201" "361,124")
        ("0,202" "361,043")
        ("0,203" "361,077")
        ("0,204" "361,114")
        ("0,205" "361,120")
        ("0,206" "361,121")
        ("0,207" "361,117")
        ("0,210" "361,122")
        ("0,211" "361,134")
        ("0,212" "361,140")
        ("0,213" "361,141")
        ("0,214" "361,145")
        ("0,215" "361,137")
        ("0,216" "361,155")
        ("0,217" "361,160")
        ("0,220" "361,142")
        ("0,221" "361,241")
        ("0,222" "361,243")
        ("0,223" "361,276")
        ("0,224" "361,250")
        ("0,225" "361,320")
        ("0,226" "361,321")
        ("0,227" "361,322")
        ("0,230" "361,322")
        ("0,231" "361,334")
        ("0,232" "361,244")
        ("0,233" "361,341")
        ("0,234" "361,261")
        ("0,235" "361,337")
        ("0,236" "361,262")
        ("0,237" "361,255")
        ("0,240" "361,247")
        ("0,244" "0,057")                                   ; Slash, but should be fraction
        ("0,246" "357,243")
        ("0,250" "0,244")
        ("0,254" "357,052")
        ("0,255" "357,053")
        ("0,256" "360,004")
        ("0,257" "360,005")
        ("0,261" EMDASH)
        ("0,262" "357,060")
        ("0,263" "357,061")
        ("0,267" "357,146")
        ("0,270" "43,262")
        ("0,271" "357,050")
        ("0,274" "41,104")
        ("0,275" "357,101")
        ("0,311" "357,153")
        ("0,314" "361,314")
        ("0,321" "375,261")
        ("0,324" "361,324")
        ("0,325" "375,362")
        ("0,326" "375,363")
        ("0,327" "0,274")
        ("0,330" "0,275")
        ("0,331" "0,264")
        ("0,332" "0,270")
        ("0,333" "357,152")
        ("0,334" "361,265")
        ("0,335" "0,261")
        ("0,336" "361,042")
        ("0,337" "357,044")
        ("0,340" "361,340")
        ("0,344" "361,041")
        ("0,345" "361,345")
```

```
        ("0,346" "361,050")
        ("0,347" "361,044")
        ("0,355" "361,355")
        ("0,356" "361,055")
        ("0,357" "361,061")
        ("0,360" "361,360")
        ("0,362" "361,062")
        ("0,364" "361,065")
        ("0,366" "361,060")
        ("0,367" "361,277")
        ("0,375" "361,100")
        ("0,376" "361,104")))

(DEFINEQ

(MCCSCODEMAPARRAY
  [LAMBDA (MAP)                                          ; 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

    ;; Atom cases for loadup

    (SELECTQ MAP
        (XCCS (SETQ MAP (APPEND MTOXCODEMAP ALTOTEXT2MCCS)))
        (MCCS (SETQ MAP ALTOTEXT2MCCS))
        NIL)
    (LET ((TABLE (ARRAY (ADD1 \MAXTHINCHAR)
                        'WORD 0 0)))
        (for I from 0 to \MAXTHINCHAR do (SETA TABLE I I))
        [for PAIR FROMCODE in MAP when (LISTP PAIR) unless (EQ '* (CAR PAIR)) when (SETQ FROMCODE
                                                                                        (CL:IF (CHARCODEP (CAR PAIR))
                                                                                               (CAR PAIR)
                                                                                               (CHARCODE.DECODE
                                                                                                (CAR PAIR)
                                                                                                T)))
           do (SETA TABLE FROMCODE (CL:IF (CHARCODEP (CADR PAIR))
                                          (CADR PAIR)
                                          (CHARCODE.DECODE (CADR PAIR))))]
        TABLE])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS ALTOTOMCCSARRAY SYMBOLTOMCCSARRAY HIPPOTOMCCSARRAY CYRILLICTOMCCSARRAY MATHTOMCCSARRAY
        SIGMATOMCCSARRAY PALATINOTOMCCSARRAY)
)

(RPAQ? ALTOTOMCCSARRAY (MCCSCODEMAPARRAY 'MCCS))

(RPAQ? SYMBOLTOMCCSARRAY (MCCSCODEMAPARRAY SYMBOLTOMCCS))

(RPAQ? HIPPOTOMCCSARRAY (MCCSCODEMAPARRAY HIPPOTOMCCS))

(RPAQ? CYRILLICTOMCCSARRAY (MCCSCODEMAPARRAY CYRILLICTOMCCS))

(RPAQ? MATHTOMCCSARRAY (MCCSCODEMAPARRAY MATHTOMCCS))

(RPAQ? SIGMATOMCCSARRAY (MCCSCODEMAPARRAY SIGMATOMCCS))

(RPAQ? PALATINOTOMCCSARRAY (MCCSCODEMAPARRAY PALATINOTOMCCS))

(DEFINEQ

(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 HIPPOTOMCODE))
        (CYRILLIC (FUNCTION CYRILLICTOMCODE))
        (XCCS (FUNCTION XTOMCODE))
        (GACHA (FUNCTION GACHATOMCODE))
        (PALATINO (FUNCTION PALATINOTOMCODE))
        (MCCS NIL)
        NIL])


(MCCSMAPPAIRS
  [LAMBDA (FROMENCODING NONIDENTITY)                    ; 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 ((FN (MCCSMAPFN FROMENCODING))
          PAIRS KEEPCS0)
         (CL:WHEN FN
             [SETQ PAIRS (SELECTQ FROMENCODING
                             (GACHA                     ; ctrl and upper are slugged
                                 [APPEND (XCCSUNDEFINEDPAIRS)
                                         '(((Uparrow TERMINAL)
                                            Circumflex)
                                           (↑X Lowline])
                             (ALTOTEXT (APPEND (XCCSUNDEFINEDPAIRS)
                                               ALTOTEXT2MCCS))
                             (XCCS$ '((Uparrow Circumflex)
                                      (Leftarrow Lowline)
                                      (Lowline Leftarrow)
                                      (Circumflex Uparrow)))
                             (PALATINO (APPEND (XCCS.CS0.UNDEFINED)
                                               PALATINOTOMCCS))
                             (PROGN (SETQ KEEPCS0 T)
                                    (for C M from 0 to \MAXTHINCHAR when (SETQ M (APPLY* FN C NONIDENTITY))
                                       collect (LIST C M]

             ;; Weed out interspersed comments, convert to charcodes

             [SETQ PAIRS (for P in PAIRS when (LISTP P) unless (EQ '* (CAR P)) collect (LIST (if (LISTP (CAR P))
                                                                                                then

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

                                                                                                    (CONS
                                                                                                     (CL:IF (CHARCODEP
                                                                                                                (CAAR P))
                                                                                                            (CAAR P)
                                                                                                            (
                                                                                                             CHARCODE.DECODE
                                                                                                                (CAAR P)))
                                                                                                     (CDAR P))
                                                                                              elseif (CHARCODEP (CAR P))
```

```
                                                                then (CAR P)
                                                              else (CHARCODE.DECODE
                                                                     (CAR P)))
                                                            (CL:IF (CHARCODEP (CADR P))
                                                                   (CADR P)
                                                                   (CHARCODE.DECODE
                                                                    (CADR P)))]
```

;; Any character that is moved gets replaced by a slug. It may then be coerced from another font. But families like SYMBOL, HIPPO ;; etc. want to preserve CS0 even if they copy their glyphs also to somewhere else.

```
[APPEND PAIRS (for P in PAIRS when (CAR P) unless [OR (AND KEEPCS0 (ILEQ (CAR P)
                                                                         \MAXTHINCHAR))
                                                      (AND (LISTP (CAR P))
                                                           (LITATOM (CADAR P)))
                                                      (thereis X in PAIRS suchthat (EQ (CADR X)
                                                                                       (CAR P]
                                  collect (LIST NIL (CAR P])])
```

```
(XCCS.CS0.UNDEFINED
  [LAMBDA NIL                                                    ; Edited  5-Oct-2025 22:44 by rmk
```

;; Maps slugs to all undefined/reserved characters in XCCS

```
    (APPEND (for I from 0 to (SUB1 (CHARCODE SPACE)) collect (LIST NIL I))
            (for I from (CHARCODE "0,#NULL") to (SUB1 (CHARCODE "0,#SPACE")) collect (LIST NIL I))
            (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 NIL I])
```

```
(XCCSUNDEFINEDPAIRS
  [LAMBDA NIL                                                    ; Edited  5-Oct-2025 22:39 by rmk
                                                                 ; Edited  2-Sep-2025 13:14 by rmk
    (APPEND (XCCS.CS0.UNDEFINED)
            (for I from 128 to \MAXTHINCHAR collect (LIST NIL I])

)
```

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

```
(DEFINEQ

(GACHATOMCODE
  [LAMBDA (GCODE)                                                ; Edited  7-Sep-2025 22:38 by rmk
                                                                 ; Edited  3-Sep-2025 23:23 by rmk
                                                                 ; Edited 30-Aug-2025 21:58 by rmk
```

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

```
    (CL:IF (EQ GCODE (CHARCODE ↑X))
           (CHARCODE Lowline)
           GCODE)])

(SYMBOLTOMCODE
  [LAMBDA (SCODE)                                                ; 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)
                 (LET ((MCODE (ELT SYMBOLTOMCCSARRAY SCODE)))
                      (CL:UNLESS (EQ MCODE SCODE)
                                 MCODE)))
        SCODE])

(SIGMATOMCODE
  [LAMBDA (SCODE)                                                ; 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)
               (LET ((MCODE (ELT SIGMATOMCCSARRAY SCODE)))
                    (CL:UNLESS (EQ MCODE SCODE)
                               MCODE)))
        SCODE])


(ATOMCODE
  [LAMBDA (ACODE)                                    ; 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)
               (LET ((MCODE (ELT ALTOTOMCCSARRAY ACODE)))
                    (CL:UNLESS (EQ MCODE ACODE)
                               MCODE)))
        ACODE])


(MATHTOMCODE
  [LAMBDA (MATHCODE)                                 ; 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)
               (LET ((MCODE (ELT MATHTOMCCSARRAY MATHCODE)))
                    (CL:UNLESS (EQ MCODE MATHCODE)
                               MCODE)))
        MATHCODE])


(HIPPOTOMCODE
  [LAMBDA (HCODE)                                    ; 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)
               (LET ((MCODE (ELT HIPPOTOMCCSARRAY HCODE)))
                    (CL:UNLESS (EQ MCODE HCODE)
                               MCODE)))
        HCODE])


(CYRILLICTOMCODE
  [LAMBDA (CCODE)                                    ; Edited  7-Sep-2025 22:40 by rmk
                                                     ; Edited 24-May-2025 09:38 by rmk
    (OR (CL:WHEN (ILEQ CCODE \MAXTHINCHAR)
               (LET ((MCODE (ELT CYRILLICTOMCCSARRAY CCODE)))
                    (CL:UNLESS (EQ MCODE CCODE)
                               MCODE)))
        CCODE])


(PALATINOTOMCODE
  [LAMBDA (PCODE)                                    ; 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)
               (LET ((MCODE (ELT PALATINOTOMCCSARRAY PCODE)))
                    (CL:UNLESS (EQ MCODE PCODE)
                               MCODE)))
        PCODE])

)
```

FUNCTION INDEX

VARIABLE INDEX

CONSTANT INDEX

MACRO INDEX