```
(RPAQQ JAPANESECOMS
       [                                                          ; XCCS to JIS converter
        [COMS                                                     ; JIS to XCCS conversion table.
              (VARS *JIS-TO-XCCS-CONV-NO-FONT-TABLE* *JIS-TO-XCCS-CODE-MAP* *HANKAKU-TO-ZENKAKU-CODE-MAP*)
              (GLOBALVARS *JIS-TO-XCCS-CONV-NO-FONT-TABLE* *JIS-TO-XCCS-CONV-TABLE-LIST* *JIS-TO-XCCS-CODE-MAP*
                      *HANKAKU-TO-ZENKAKU-CODE-MAP* *JIS-1KU-TO-XCCS-CONV-TABLE* *JIS-2KU-TO-XCCS-CONV-TABLE*
                      *JIS-6KU-TO-XCCS-CONV-TABLE* *XCCS-TO-JIS-CONV-TABLE* *HANKAKU-TO-ZENKAKU-CONV-TABLE*
                      *ZENKAKU-TO-HANKAKU-CONV-TABLE*)
              (FNS \MAKE.JIS.TO.XCCS.CONV.TABLE)
              (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\MAKE.JIS.TO.XCCS.CONV.TABLE]
        (DECLARE%: DOEVAL@COMPILE DONTCOPY (MACROS \CONV.JIS.TO.XCCS \DO.CONV.JIS.TO.XCCS))
        (FNS \JISIN \JISPEEK \BACKJISCCODE \SHIFTJISIN \SHIFTJISPEEK \BACKSHIFTJISCCODE \EUCIN \EUCPEEK
              \BACKEUCCODE)
        (FNS \JISOUTCHARFN \SHIFTJISOUTCHARFN \EUCOUTCHARFN)
        (COMS (FNS CONVHANKAKU)
              (DECLARE%: DOEVAL@COMPILE DONTCOPY (MACROS \CONV.XCCS.TO.JIS \DO.CONV.XCCS.TO.JIS \ASCIIP
                                                         \NOT.EQUIVALENT.TO.JIS \CONV.HANKAKU.TO.ZENKAKUP
                                                         \CONV.ZENKAKU.KANA)))
        (DECLARE%: DOEVAL@COMPILE DONTCOPY

                 ;; JIS specific macro

                 (MACROS \EXTRACT.NO.FONT.CODE \EXTARACT.CONV.TABLE \NOT.EQUIVALENT.TO.XCCS \EXTRACT.SET
                         \EXTRACT.CODE \CHNAGE.KI.MODE \KIMODEP \HANKAKUP \KANJIP \NOTGAIJIP \INVALID.TENP
                         \CONV.HANKAKU.KANA \OUTKI \OUTKO)

                 ;; Shift-JIS specific macro

                 (MACROS \CONV.SJIS.TO.JIS \CONV.JIS.TO.SJIS \SJIS.KANJI.FIRST.BYTEP)

                 ;; EUC specific macro

                 (MACROS \EUC.KANJI.FIRST.BYTEP \GAIJIP \EUC.HANKAKUP))
        (FNS \CREATE.JIS.EXTERNALFORMAT \CREATE.SHIFTJIS.EXTERNALFORMAT \CREATE.EUC.EXTERNALFORMAT)
        (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\CREATE.JIS.EXTERNALFORMAT)
                                          (\CREATE.SHIFTJIS.EXTERNALFORMAT)
                                          (\CREATE.EUC.EXTERNALFORMAT)))
        [DECLARE%: EVAL@COMPILE DONTCOPY (P (OR (GETP 'EXPORTS.ALL 'FILE)
                                                (PRINT "NOTE: JAPANESE requires EXPORTS.ALL for compilation" T]
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                              (NLAML)
                                                                              (LAMA CONVHANKAKU])
```

;; XCCS to JIS converter

;; JIS to XCCS conversion table.

```
(RPAQQ *JIS-TO-XCCS-CONV-NO-FONT-TABLE*
       ((8484 . 8484)
        (8485 . 8485)
        (8497 . 9155)
        (8798 . 61376)
        (8802 . 8802)
        (8805 . 64892)
        (8806 . 64894)
        (8820 . 9148)
        (8821 . 9132)
        (8822 . 213)
        (8830 . 8830)
        (10273 . 61189)
        (10274 . 61188)
        (10275 . 10275)
        (10276 . 10276)
        (10277 . 10277)
        (10278 . 10278)
        (10279 . 10279)
        (10280 . 10280)
        (10281 . 10281)
        (10282 . 10282)
        (10283 . 61414)
        (10284 . 61410)
        (10285 . 61409)
        (10286 . 10286)
        (10287 . 10287)
        (10288 . 10288)
        (10289 . 10289)
        (10290 . 10290)
        (10291 . 10291)
        (10292 . 10292)
```

```
                (10293 . 10293)
                (10294 . 61411)
                (10295 . 10295)
                (10296 . 10296)
                (10297 . 10297)
                (10298 . 10298)
                (10299 . 10299)
                (10300 . 10300)
                (10301 . 10301)
                (10302 . 10302)
                (10303 . 10303)
                (10304 . 10304)))

(RPAQQ *JIS-TO-XCCS-CODE-MAP*
        ((1  (1 33 .  33)
             (2 33 .  34)
             (3 33 .  35)
             (6 0 . 183)
             (7 0 .  58)
             (8 0 .  59)
             (9 0 .  63)
             (10 0 .  33)
             (11 33 .  43)
             (12 33 .  44)
             (13 0 . 194)
             (14 0 . 193)
             (15 0 . 200)
             (16 0 . 195)
             (18 0 . 204)
             (19 33 .  51)
             (20 33 .  52)
             (21 33 .  53)
             (22 33 .  54)
             (23 33 .  55)
             (24 33 .  56)
             (25 33 .  57)
             (26 33 .  58)
             (27 33 .  59)
             (28 33 .  60)
             (29 239 .  36)
             (30 33 .  62)
             (31 0 .  47)
             (32 0 .  92)
             (33 0 . 126)
             (34 33 .  66)
             (35 0 . 124)
             (36 33 .  68)
             (37 33 .  69)
             (38 0 . 169)
             (39 0 .  39)
             (40 0 . 170)
             (41 0 . 186)
             (42 0 .  40)
             (43 0 .  41)
             (44 33 .  76)
             (45 33 .  77)
             (46 0 .  91)
             (47 0 .  93)
             (48 0 . 123)
             (49 0 . 125)
             (50 239 .  50)
             (51 239 .  51)
             (52 0 . 171)
             (53 0 . 187)
             (54 33 .  86)
             (55 33 .  87)
             (56 33 .  88)
             (57 33 .  89)
             (58 33 .  90)
             (59 33 .  91)
             (60 0 .  43)
             (61 0 .  45)
             (62 0 . 177)
             (63 0 . 180)
             (64 0 . 184)
             (65 0 .  61)
             (66 33 .  98)
             (67 0 .  60)
             (68 0 .  62)
             (69 33 . 101)
             (70 33 . 102)
             (71 33 . 103)
             (72 33 . 104)
             (73 33 . 105)
             (74 33 . 106)
             (75 0 . 176)
             (76 33 . 108)
```

```
                    (77 33 . 109)
                    (78 33 . 110)
                    (79 0 . 165)
                    (80 0 . 164)
                    (81 0 . 162)
                    (82 0 . 163)
                    (83 0 . 37)
                    (84 0 . 35)
                    (85 0 . 38)
                    (86 0 . 42)
                    (87 0 . 64)
                    (88 0 . 167)
                    (89 33 . 121)
                    (90 33 . 122)
                    (91 33 . 123)
                    (92 33 . 124)
                    (93 33 . 125)
                    (94 33 . 126))
              (2 (1 34 . 33)
                    (2 34 . 34)
                    (3 34 . 35)
                    (4 34 . 36)
                    (5 34 . 37)
                    (6 34 . 38)
                    (7 34 . 39)
                    (8 34 . 40)
                    (9 34 . 41)
                    (10 0 . 174)
                    (11 0 . 172)
                    (12 0 . 173)
                    (13 0 . 175)
                    (14 34 . 46)
                    (26 239 . 74)
                    (27 239 . 76)
                    (28 239 . 89)
                    (29 239 . 88)
                    (30 239 . 91)
                    (31 239 . 90)
                    (32 239 . 87)
                    (33 239 . 86)
                    (42 239 . 182)
                    (43 239 . 183)
                    (44 239 . 106)
                    (45 239 . 79)
                    (46 239 . 78)
                    (47 239 . 181)
                    (48 239 . 180)
                    (60 239 . 108)
                    (61 239 . 112)
                    (63 239 . 186)
                    (64 239 . 185)
                    (65 239 . 114)
                    (67 239 . 66)
                    (68 239 . 67)
                    (71 239 . 113)
                    (72 239 . 111)
                    (73 239 . 117)
                    (74 34 . 106)
                    (82 241 . 40)
                    (83 239 . 65)
                    (87 239 . 48)
                    (88 239 . 49)
                    (89 0 . 176))
              (6 (1 38 . 65)
                    (2 38 . 66)
                    (3 38 . 68)
                    (4 38 . 69)
                    (5 38 . 70)
                    (6 38 . 73)
                    (7 38 . 74)
                    (8 38 . 75)
                    (9 38 . 76)
                    (10 38 . 77)
                    (11 38 . 78)
                    (12 38 . 79)
                    (13 38 . 80)
                    (14 38 . 81)
                    (15 38 . 82)
                    (16 38 . 83)
                    (17 38 . 85)
                    (18 38 . 86)
                    (19 38 . 88)
                    (20 38 . 89)
                    (21 38 . 90)
                    (22 38 . 91)
                    (23 38 . 92)
                    (24 38 . 93)
```

```
                    (33 38 . 97)
                    (34 38 . 98)
                    (35 38 . 100)
                    (36 38 . 101)
                    (37 38 . 102)
                    (38 38 . 105)
                    (39 38 . 106)
                    (40 38 . 107)
                    (41 38 . 108)
                    (42 38 . 109)
                    (43 38 . 110)
                    (44 38 . 111)
                    (45 38 . 112)
                    (46 38 . 113)
                    (47 38 . 114)
                    (48 38 . 115)
                    (49 38 . 117)
                    (50 38 . 118)
                    (51 38 . 120)
                    (52 38 . 121)
                    (53 38 . 122)
                    (54 38 . 123)
                    (55 38 . 124)
                    (56 38 . 125)))))

(RPAQQ *HANKAKU-TO-ZENKAKU-CODE-MAP*
          ((161 . 8483)
           (162 . 8534)
           (163 . 8535)
           (164 . 8482)
           (165 . 183)
           (166 . 9586)
           (167 . 9505)
           (168 . 9507)
           (169 . 9509)
           (170 . 9511)
           (171 . 9513)
           (172 . 9571)
           (173 . 9573)
           (174 . 9575)
           (175 . 9539)
           (176 . 8508)
           (177 . 9506)
           (178 . 9508)
           (179 . 9510)
           (180 . 9512)
           (181 . 9514)
           (182 . 9515)
           (183 . 9517)
           (184 . 9519)
           (185 . 9521)
           (186 . 9523)
           (187 . 9525)
           (188 . 9527)
           (189 . 9529)
           (190 . 9531)
           (191 . 9533)
           (192 . 9535)
           (193 . 9537)
           (194 . 9540)
           (195 . 9542)
           (196 . 9544)
           (197 . 9546)
           (198 . 9547)
           (199 . 9548)
           (200 . 9549)
           (201 . 9550)
           (202 . 9551)
           (203 . 9554)
           (204 . 9557)
           (205 . 9560)
           (206 . 9563)
           (207 . 9566)
           (208 . 9567)
           (209 . 9568)
           (210 . 9569)
           (211 . 9570)
           (212 . 9572)
           (213 . 9574)
           (214 . 9576)
           (215 . 9577)
           (216 . 9578)
           (217 . 9579)
           (218 . 9580)
           (219 . 9581)
           (220 . 9583)
           (221 . 9587)
```

```
                    (222 . 8491)
                    (223 . 8492)))

(DECLARE%: DOEVAL@COMPILE DONTCOPY


(GLOBALVARS *JIS-TO-XCCS-CONV-NO-FONT-TABLE* *JIS-TO-XCCS-CONV-TABLE-LIST* *JIS-TO-XCCS-CODE-MAP*
        *HANKAKU-TO-ZENKAKU-CODE-MAP* *JIS-1KU-TO-XCCS-CONV-TABLE* *JIS-2KU-TO-XCCS-CONV-TABLE*
        *JIS-6KU-TO-XCCS-CONV-TABLE* *XCCS-TO-JIS-CONV-TABLE* *HANKAKU-TO-ZENKAKU-CONV-TABLE*
        *ZENKAKU-TO-HANKAKU-CONV-TABLE*)
)

(DEFINEQ
```

(\MAKE.JIS.TO.XCCS.CONV.TABLE

```
  [LAMBDA NIL                                                              ; Edited 20-Feb-91 19:28 by nm
```

;;; The JIS codes which are not equivalent to XCCS reside in 1, 2, 3, 6, 8 and 84 KU. In case of 3 and 84 KU, the corresponding XCCS is calicutated
;;; from JIS. In case of 1,2 and 6 KU, we have to prepare conversion tables for each because the mapping between XCCS and JIS are random. 8 KU is
;;; treated specially because no displayable font is assigned for 8 KU in XCCS. They are handled with *JIS-TO-XCCS-CONV-NO-FONT-TABLE*.

;;; Each conversion table is an byte array of size 188 (94 * 2). 94 is a largest number of TEN. TEN is one origin. Each JIS code is represented with two
;;; bytes in the table. The first byte is a character set and the second byte is a character code in XCCS. If both of the first byte and the second byte are
;;; 255, it means the JIS code is not defined for the entry. If the first byte is 255 and the second byte is 0, it means a JIS code is defined for the entry and
;;; there is a XCCS code corresponding to the JIS code, but no displayable font is assigned for the code in XCCS. In the last case, the real XCCS code
;;; is found in *JIS-TO-XCCS-CONV-NO-FONT-TABLE*.

;;; *HANKAKU-TO-ZENKAKU-CONV-TABLE* holds the mapping between JIS HANAKAKU-KANA code to XCCS. XCCS does not support HANKAKU
;;; code.

```
    (SETQ *JIS-1KU-TO-XCCS-CONV-TABLE* (ARRAY 188 'BYTE 255))
    (SETQ *JIS-2KU-TO-XCCS-CONV-TABLE* (ARRAY 188 'BYTE 255))
    (SETQ *JIS-6KU-TO-XCCS-CONV-TABLE* (ARRAY 188 'BYTE 255))
    (SETQ *XCCS-TO-JIS-CONV-TABLE* (HASHARRAY 256))
    (SETQ *HANKAKU-TO-ZENKAKU-CONV-TABLE* (HASHARRAY 64))
    (SETQ *ZENKAKU-TO-HANKAKU-CONV-TABLE* (HASHARRAY 64))
    (CL:DO ((TABLES (LIST *JIS-1KU-TO-XCCS-CONV-TABLE* *JIS-2KU-TO-XCCS-CONV-TABLE* *JIS-6KU-TO-XCCS-CONV-TABLE*
                         )
                   (CDR TABLES))
            (KU '(1 2 6)
                (CDR KU))
            CODEMAP)
           ((CL:ENDP TABLES))
        (SETQ CODEMAP (CDR (ASSOC (CAR KU)
                                  *JIS-TO-XCCS-CODE-MAP*)))
        (for MAP in CODEMAP do (SETA (CAR TABLES)
                                     (IDIFFERENCE (UNFOLD (CAR MAP)
                                                          2)
                                                  1)
                                     (CADR MAP))
                              (SETA (CAR TABLES)
                                    (UNFOLD (CAR MAP)
                                            2)
                                    (CDDR MAP))))
    (bind KU TEN TABLE for ENTRY in *JIS-TO-XCCS-CONV-NO-FONT-TABLE*
       do (SETQ KU (IDIFFERENCE (FOLDLO (CAR ENTRY)
                                        256)
                                32))
          (SETQ TABLE (SELECTQ KU
                         (1 *JIS-1KU-TO-XCCS-CONV-TABLE*)
                         (2 *JIS-2KU-TO-XCCS-CONV-TABLE*)
                         (6 *JIS-6KU-TO-XCCS-CONV-TABLE*)
                         NIL))
          (AND TABLE (SETA TABLE (UNFOLD (IDIFFERENCE (LOGAND 255 (CAR ENTRY))
                                                      32)
                                         2)
                                 0)))
    (for MAP in *HANKAKU-TO-ZENKAKU-CODE-MAP* do (PUTHASH (CAR MAP)
                                                         (CDR MAP)
                                                         *HANKAKU-TO-ZENKAKU-CONV-TABLE*))
    (for MAP in *HANKAKU-TO-ZENKAKU-CODE-MAP* do (PUTHASH (CDR MAP)
                                                         (CAR MAP)
                                                         *ZENKAKU-TO-HANKAKU-CONV-TABLE*))
    (for MAP in (APPEND [for KU in *JIS-TO-XCCS-CODE-MAP*
                           join (for TEN in (CDR KU) collect '(,(LOGOR (UNFOLD (CADR TEN)
                                                                              256)
                                                                       (CDDR TEN))
                                                              \,
                                                              (LOGOR (UNFOLD (IPLUS (CAR KU)
                                                                                    32)
                                                                             256)
                                                                     (IPLUS (CAR TEN)
                                                                            32]
                       *JIS-TO-XCCS-CONV-NO-FONT-TABLE*)
       do (PUTHASH (CAR MAP)
                   (CDR MAP)
                   *XCCS-TO-JIS-CONV-TABLE*))
    (SETQ *JIS-TO-XCCS-CONV-TABLE-LIST* '((33 \, *JIS-1KU-TO-XCCS-CONV-TABLE*)
```

```
                                                    (34 \, *JIS-2KU-TO-XCCS-CONV-TABLE*)
                                                    (38 \, *JIS-6KU-TO-XCCS-CONV-TABLE*])
)

(DECLARE%: DONTEVAL@LOAD DOCOPY
```

(**MAKE.JIS.TO.XCCS.CONV.TABLE**)
```
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS \CONV.JIS.TO.XCCS  MACRO [OPENLAMBDA (KU TEN)
```

;;;; Some character code is not equivalent between JIS and XCCS. In such case, we have to convert the character to corresponding XCCS.

```
                              (COND
                                ((\NOT.EQUIVALENT.TO.XCCS KU)
                                 (\DO.CONV.JIS.TO.XCCS KU TEN))
                                (T (LOGOR (UNFOLD KU 256)
                                          TEN])
```

(PUTPROPS **DO.CONV.JIS.TO.XCCS** MACRO
```
            [(KU TEN)
```

;;; Convert a JIS code divided into KU (high 8 bit) and TEN (low 8 bit) to an corresponding XCCS code.

```
            (COND
              ((\INVALID.TENP TEN)
               *DEFAULT-NOT-CONVERTED-FAT-CODE*)
              (T (SELECTQ KU
                   ((33 34 38)                                              ; 1, 2 and 6 KU
                     [LET* ((CONVTABLE (\EXTARACT.CONV.TABLE KU))
                            (SET (\EXTRACT.SET TEN CONVTABLE))
                            (CODE (\EXTRACT.CODE TEN CONVTABLE)))
                           (COND
                             ((NEQ SET 255)
                              (LOGOR (UNFOLD SET 256)
                                     CODE))
                             (T (COND
                                  ((EQ CODE 255)                  ; Not defined in JIS.
                                   *DEFAULT-NOT-CONVERTED-FAT-CODE*)
                                  (T                              ; Defined in JIS but the displayable font is not assigned in the
                                                                  ; corresponding code in XCCS.
                                     (COND
                                       (*REPLACE-NO-FONT-CODE* *DEFAULT-NOT-CONVERTED-FAT-CODE*)
                                       (T (\EXTRACT.NO.FONT.CODE (LOGOR (UNFOLD KU 256)
                                                                        TEN])
                   (35                                           ; 3 KU
                                                                 ; Alpha numeric codes are all defined as single byte codes in
                                                                 ; XCCS.
                      TEN)
                   (40                                           ; 8 KU
                      (COND
                        [(< 0 TEN 33)
                         (COND
                           (*REPLACE-NO-FONT-CODE* *DEFAULT-NOT-CONVERTED-FAT-CODE*)
                           (T (\EXTRACT.NO.FONT.CODE (LOGOR KU TEN]
                        (T *DEFAULT-NOT-CONVERTED-FAT-CODE*)))
                   (116                                          ; 84 KU
                      (COND
                        ((< 0 TEN 5)
                         (LOGOR 29952 TEN))
                        (T *DEFAULT-NOT-CONVERTED-FAT-CODE*)))
                   (117                                          ; 85 KU
                      (COND
                        ((< 0 TEN 28)
                         (LOGOR 29696 TEN))
                        (T *DEFAULT-NOT-CONVERTED-FAT-CODE*)))
                   *DEFAULT-NOT-CONVERTED-FAT-CODE*])
)
)

(DEFINEQ
```

(**JISIN**
```
  [LAMBDA (STREAM COUNTP)                                        ; Edited  6-Aug-2021 17:07 by rmk:
```

;;;; Returns a 16 bit XCCS code. Assuming the input character stream is encoded with JIS. Allows the incorrect usage of KI and KO based on the two
;;;; different JIS, OLDJIS and NEWJIS, because it is very likely that these two different sets of KI and KO are used simultaneously, although it is against a
;;;; standard!

;;; If COUNTP is non-NIL, the variable *BYTECOUNTER* is set freely to the number of bytes read. Doesn't do EOL conversion -- \INCCODE do that.

```
    (DECLARE (USEDFREE *BYTECOUNTER*))
```

```
    (PROG (CH1 CH2 CH3 (IN16BITFLG (\KIMODEP STREAM T))
              (CHARNUM 0))
     RETRY
        (CL:WHEN (SETQ CH1 (\BIN STREAM))
          (IF (EQ CH1 (CHARCODE ESC))
              THEN                                              ; Might be KI or KO.
                    (SETQ CH2 (\BIN STREAM))
                    (IF (EQ CH2 (CHARCODE $))
                        THEN                                    ; Might be KI
                            (SETQ CH3 (\BIN STREAM))
                            [IF (OR (EQ CH3 (CHARCODE B))
                                    (EQ CH3 (CHARCODE @)))
                                THEN                            ; KI
                                    (\CHNAGE.KI.MODE STREAM T T)
                                    (CL:WHEN COUNTP (ADD CHARNUM 3))
                                                ; Here we have to try the same preocedure again, because
                                                ; bogus duplicated KI/KO sequence might come again!
                                    (SETQ IN16BITFLG T)
                                    (GO RETRY)
                                ELSE (RETURN (IF IN16BITFLG
                                                 THEN           ; Under processing 16 bit code.
                                                     (\BACKFILEPTR STREAM)
                                                     (CL:WHEN COUNTP
                                                        (SETQ *BYTECOUNTER* (IPLUS 2 CHARNUM)))
                                                     (\CONV.JIS.TO.XCCS CH1 CH2)
                                                 ELSE (\BACKFILEPTR STREAM)
                                                     (\BACKFILEPTR STREAM)
                                                     (CL:WHEN COUNTP
                                                        (SETQ *BYTECOUNTER* (IPLUS 1 CHARNUM)))
                                                     (CHARCODE ESC]
                        ELSEIF (EQ CH2 (CHARCODE %())
                          THEN                                  ; Might be KO
                            (SETQ CH3 (\BIN STREAM))
                            [IF (OR (EQ CH3 (CHARCODE J))
                                    (EQ CH3 (CHARCODE H)))
                                THEN                            ; KO
                                    (\CHNAGE.KI.MODE STREAM T NIL)
                                    (CL:WHEN COUNTP (ADD CHARNUM 3))
                                                ; Oops.  Yes, we have to try again to ignore duplicated KI/KO
                                                ; sequence.
                                    (SETQ IN16BITFLG NIL)
                                    (GO RETRY)
                                ELSE (RETURN (IF IN16BITFLG
                                                 THEN           ; Under processing 16 bit code.
                                                     (\BACKFILEPTR STREAM)
                                                     (CL:WHEN COUNTP
                                                        (SETQ *BYTECOUNTER* (IPLUS CHARNUM 2)))
                                                     (\CONV.JIS.TO.XCCS CH1 CH2)
                                                 ELSE (\BACKFILEPTR STREAM)
                                                     (\BACKFILEPTR STREAM)
                                                     (CL:WHEN COUNTP
                                                        (SETQ *BYTECOUNTER* (IPLUS CHARNUM 1)))
                                                     (CHARCODE ESC]
                        ELSEIF IN16BITFLG
                          THEN                                  ; Under processing 16 bit code.
                            (CL:WHEN COUNTP
                                (SETQ *BYTECOUNTER* (IPLUS CHARNUM 2)))
                            (RETURN (\CONV.JIS.TO.XCCS CH1 CH2))
                        ELSE (\BACKFILEPTR STREAM)
                            (CL:WHEN COUNTP
                                (SETQ *BYTECOUNTER* (IPLUS CHARNUM 1)))
                            (RETURN (CHARCODE ESC)))
              ELSEIF IN16BITFLG
                THEN                                            ; Under processing 16 bit code.
                    (CL:WHEN COUNTP
                        (SETQ *BYTECOUNTER* (IPLUS CHARNUM 2)))
                    (RETURN (\CONV.JIS.TO.XCCS CH1 (\BIN STREAM)))
              ELSEIF (\HANKAKUP CH1)
                THEN                                            ; HANKAKU-KATAKANA is converted to ZENKAKU-KATAKANA
                                                                ; because XCCS does not support HANKAKU-KATAKANA.
                    (CL:WHEN COUNTP
                        (SETQ *BYTECOUNTER* (IPLUS CHARNUM 1)))
                    (RETURN (\CONV.HANKAKU.KANA CH1))
              ELSE ;; C0, SP, DEL, C1, 10/0, or 15/15 of 0 character set.

                    (CL:WHEN COUNTP
                        (SETQ *BYTECOUNTER* (IPLUS CHARNUM 1)))
                    (RETURN CH1))])])
```

\JISPEEK
```
  [LAMBDA (STREAM NOERROR COUNTP)                                       ; Edited 25-Feb-91 16:27 by nm
```

;;; Returns a 16 bit XCCS code. Assuming the input character stream is encoded with JIS. Allows the incorrect usage of KI and KO based on the two
;;; different JIS, OLDJIS and NEWJIS, because it is very likely that these two different sets of KI and KO are used simultaneously, although it is against a
;;; standard! May actually read the KI or KO.

```
;;; If COUNTP is non-NIL, the number of bytes read is returned as a second value.  Doesn't do EOL conversion -- \INCHAR or \INCCODE do that.

    (PROG ((IN16BITFLG (\KIMODEP STREAM T))
           (CHARNUM 0)
           (CH1 (\PEEKBIN STREAM NOERROR))
           CH2 CH3)
      RETRY
          (COND
            [(NULL CH1)
              (COND
                (COUNTP (RETURN (CL:VALUES NIL CHARNUM)))
                (T (RETURN NIL]
            [(EQ CH1 (CHARCODE ESC))                              ; Might be KI or KO.
             (\BIN STREAM)                                        ; Consume the first ESC.
             (SETQ CH2 (\PEEKBIN STREAM NOERROR))
             (COND
               [(NULL CH2)
                (\BACKFILEPTR STREAM)
                (COND
                  (COUNTP (RETURN (CL:VALUES NIL CHARNUM)))
                  (T (RETURN NIL]
               [(EQ CH2 (CHARCODE $))                             ; Might be KI
                (\BIN STREAM)                                     ; Consume the $.
                (SETQ CH3 (\PEEKBIN STREAM NOERROR))
                (COND
                  [(NULL CH3)
                   (\BACKFILEPTR STREAM)
                   (\BACKFILEPTR STREAM)
                   (COND
                     (COUNTP (RETURN (CL:VALUES NIL CHARNUM)))
                     (T (RETURN NIL]
                  ((OR (EQ CH3 (CHARCODE B))
                       (EQ CH3 (CHARCODE @)))                     ; KI
                   (\CHNAGE.KI.MODE STREAM T T)
                   (AND COUNTP (SETQ CHARNUM (IPLUS CHARNUM 3)))
                   (\BIN STREAM)                                  ; Consume the B or @.
                   (SETQ IN16BITFLG T)
                   (GO RETRY))
                  (T (\BACKFILEPTR STREAM)
                     (\BACKFILEPTR STREAM)
                     (COND
                       [IN16BITFLG                                ; Under processing 16 bit code.
                         (COND
                           (COUNTP (RETURN (CL:VALUES (\CONV.JIS.TO.XCCS CH1 CH2)
                                                       CHARNUM)))
                           (T (RETURN (\CONV.JIS.TO.XCCS CH1 CH2]
                       (T (COND
                            (COUNTP (RETURN (CL:VALUES (CHARCODE ESC)
                                                        CHARNUM)))
                            (T (RETURN (CHARCODE ESC]
               [(EQ CH2 (CHARCODE %())                            ; Might be KO
                (\BIN STREAM)                                     ; Consume the (.
                (SETQ CH3 (\PEEKBIN STREAM NOERROR))
                (COND
                  [(NULL CH3)
                   (\BACKFILEPTR STREAM)
                   (\BACKFILEPTR STREAM)
                   (COND
                     (COUNTP (RETURN (CL:VALUES NIL CHARNUM)))
                     (T (RETURN NIL]
                  ((OR (EQ CH3 (CHARCODE J))
                       (EQ CH3 (CHARCODE H)))                     ; KO
                   (\CHNAGE.KI.MODE STREAM T NIL)
                   (AND COUNTP (SETQ CHARNUM 3))
                   (\BIN STREAM)                                  ; Consume the J or H.
                   (SETQ IN16BITFLG NIL)
                   (GO RETRY))
                  (T (\BACKFILEPTR STREAM)
                     (\BACKFILEPTR STREAM)
                     (COND
                       [IN16BITFLG                                ; Under processing 16 bit code.
                         (COND
                           (COUNTP (RETURN (CL:VALUES (\CONV.JIS.TO.XCCS CH1 CH2)
                                                       CHARNUM)))
                           (T (RETURN (\CONV.JIS.TO.XCCS CH1 CH2]
                       (T (COND
                            (COUNTP (RETURN (CL:VALUES (CHARCODE ESC)
                                                        CHARNUM)))
                            (T (RETURN (CHARCODE ESC]
               [IN16BITFLG                                        ; Under processing 16 bit code.
                 (\BACKFILEPTR STREAM)
                 (COND
                   (COUNTP (RETURN (CL:VALUES (\CONV.JIS.TO.XCCS CH1 CH2)
                                               CHARNUM)))
                   (T (RETURN (\CONV.JIS.TO.XCCS CH1 CH2]
               (T (\BACKFILEPTR STREAM)
                  (COND
```

```
                        (COUNTP (RETURN (CL:VALUES (CHARCODE ESC)
                                                     CHARNUM)))
                        (T (RETURN (CHARCODE ESC]
            [IN16BITFLG                                              ; Under processing 16 bit code.
                (\BIN STREAM)                                        ; Consume the first byte.
                (SETQ CH2 (\PEEKBIN STREAM NOERROR))
                (\BACKFILEPTR STREAM)
                (COND
                   [CH2 (COND
                           (COUNTP (RETURN (CL:VALUES (\CONV.JIS.TO.XCCS CH1 CH2)
                                                        CHARNUM)))
                           (T (RETURN (\CONV.JIS.TO.XCCS CH1 CH2]
                   (T (COND
                         (COUNTP (RETURN (CL:VALUES NIL CHARNUM)))
                         (T (RETURN NIL]
            [(\HANKAKUP CH1)                                         ; HANKAKU-KATAKANA is converted to ZENKAKU-KATAKANA
                                                                     ; because XCCS does not support HANKAKU-KATAKANA.
              (COND
                 (COUNTP (RETURN (CL:VALUES (\CONV.HANKAKU.KANA CH1)
                                              CHARNUM)))
                 (T (RETURN (\CONV.HANKAKU.KANA CH1]
            (T  ;; C0, SP, DEL, C1, 10/0, or 15/15 of 0 character set.

                (COND
                   (COUNTP (RETURN (CL:VALUES CH1 CHARNUM)))
                   (T (RETURN CH1])
```

(**BACKJISCCODE**
```
  [LAMBDA (STREAM COUNTP)                                            ; Edited  6-Aug-2021 17:01 by rmk:
    (DECLARE (USEDFREE *BYTECOUNTER*))
    (CL:WHEN (\BACKFILEPTR STREAM)
        (IF (\KIMODEP STREAM T)
            THEN (IF (\BACKFILEPTR STREAM)
                     THEN (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -2))
                   ELSEIF COUNTP
                     THEN (SETQ *BYTECOUNTER* -1))
          ELSEIF COUNTP
            THEN (SETQ *BYTECOUNTER* -1)))])
```

(**SHIFTJISIN**
```
  [LAMBDA (STREAM COUNTP)                                            ; Edited  6-Aug-2021 16:27 by rmk:
```
;;; Returns a 16 bit XCCS code. Assuming the input character stream is encoded with Shift-JIS. If COUNTP is non-NIL, the variable *BYTECOUNTER*
;;; is set freely to the number of byte.. Doesn't do EOL conversion -- \INCCODE.EOLC does that..

```
    (DECLARE (USEDFREE *BYTECOUNTER*))
    (LET ((CH1 (\BIN STREAM))
          CH2)
         (CL:WHEN CH1
            [COND
               ((\SJIS.KANJI.FIRST.BYTEP CH1)                        ; Read next byte and compose a kanji character.
                (\CONV.SJIS.TO.JIS CH1 (\BIN STREAM))                ; CH1 and CH2 is adjusted to represent JIS code in
                                                                     ; \CONV.SJIS.TO.JIS.
                (CL:WHEN COUNTP (SETQ *BYTECOUNTER* 2))
                (\CONV.JIS.TO.XCCS CH1 CH2))
               (T                                                    ; ASCII or HANKAKU-KATAKANA
                 (CL:WHEN COUNTP (SETQ *BYTECOUNTER* 1))
                 (COND
                    ((\HANKAKUP CH1)                                 ; HANKAKU-KATAKANA
                     (\CONV.HANKAKU.KANA CH1))
                    (T                                               ; ASCII
                       CH1])])
```

(**SHIFTJISPEEK**
```
  [LAMBDA (STREAM NOERROR COUNTP)                                    ; Edited 25-Feb-91 16:30 by nm
```
;;; Returns a 16 bit XCCS code. Assuming the input character stream is encoded with Shift-JIS. Doesn't do EOL conversion -- \INCHAR or \INCCODE
;;; do that.

```
    (PROG ((CH1 (\PEEKBIN STREAM NOERROR))
           CH2)
          (COND
             [(NULL CH1)
              (COND
                 (COUNTP (RETURN (CL:VALUES NIL 0)))
                 (T (RETURN NIL]
             [(\SJIS.KANJI.FIRST.BYTEP CH1)                          ; Read next byte and compose a kanji character.
              (\BIN STREAM)                                          ; Consume the first byte.
              [COND
                 ((NULL (SETQ CH2 (\PEEKBIN STREAM NOERROR)))
                  (\BACKFILEPTR STREAM)
                  (COND
                     (COUNTP (RETURN (CL:VALUES NIL 0)))
                     (T (RETURN NIL]
```

```
                    (\BACKFILEPTR STREAM)
                    (\CONV.SJIS.TO.JIS CH1 CH2)                                      ; CH1 and CH2 is adjusted to represent JIS code in
                                                                                    ; \CONV.SJIS.TO.JIS.
                    (COND
                       (COUNTP (RETURN (CL:VALUES (\CONV.JIS.TO.XCCS CH1 CH2)
                                                  0)))
                       (T (RETURN (\CONV.JIS.TO.XCCS CH1 CH2]
                (T                                                                   ; ASCII or HANKAKU-KATAKANA
                  (RETURN (COND
                             [(\HANKAKUP CH1)                                        ; HANKAKU-KATAKANA
                               (COND
                                  (COUNTP (RETURN (CL:VALUES (\CONV.HANKAKU.KANA CH1)
                                                             0)))
                                  (T (RETURN (\CONV.HANKAKU.KANA CH1]
                             (T                                                      ; ASCII
                               (COND
                                  (COUNTP (RETURN (CL:VALUES CH1 0)))
                                  (T (RETURN CH1])
```

(**BACKSHIFTJISCCODE**

```
  [LAMBDA (STREAM COUNTP)                                                            ; Edited  6-Aug-2021 16:32 by rmk:
    (DECLARE (USEDFREE *BYTECOUNTER*))
    (CL:WHEN (\BACKFILEPTR STREAM)
        (IF (\BACKFILEPTR STREAM)
            THEN (IF (\SJIS.KANJI.FIRST.BYTEP (\PEEKBIN STREAM))
                     THEN (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -2))
                   ELSE (\BIN STREAM)
                        (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -1)))
          ELSEIF COUNTP
            THEN (SETQ *BYTECOUNTER* -1)))])
```

(**EUCIN**

```
  [LAMBDA (STREAM COUNTP)                                                            ; Edited  6-Aug-2021 16:36 by rmk:
```

;;; Returns a 16 bit XCCS code. Assuming the input character stream is encoded with EUC (Extended Unix Codes). Although EUC is independent of a
;;; particular language, the language implemented here is Japanese, thus this should be called as UJIS (Unixnized extended JIS code). JEIDA uses
;;; EUC as UJIS.

;;; If COUNTP is non-NIL, the variable *BYTECOUNTER* is set freely to the number of bytes read. Doesn't do EOL conversion -- \INCCODE.EOLC
;;; does that.

```
    (DECLARE (USEDFREE *BYTECOUNTER*))
    (LET ((CH1 (\BIN STREAM))
          CH2)
         (CL:WHEN CH1
             (COND
                ((\EUC.KANJI.FIRST.BYTEP CH1)                                        ; Read next byte and compose a kanji character.
                 (CL:WHEN COUNTP (SETQ *BYTECOUNTER* 2))
                 (\CONV.JIS.TO.XCCS (LOGAND CH1 127)
                        (LOGAND (\BIN STREAM)
                                127)))
                ((\EUC.HANKAKUP CH1)
                 (CL:WHEN COUNTP (SETQ *BYTECOUNTER* 2))
                 (\CONV.HANKAKU.KANA (\BIN STREAM)))
                ((\GAIJIP CH1)
                 (CL:WHEN COUNTP (SETQ *BYTECOUNTER* 3))
                 (\CONV.JIS.TO.XCCS (LOGAND (\BIN STREAM)
                                            127)
                        (LOGAND (\BIN STREAM)
                                127)))
                (T                                                                   ; ASCII, C0, C1, SP or DEL
                   (CL:WHEN COUNTP (SETQ *BYTECOUNTER* 1))
                   CH1)))])
```

(**EUCPEEK**

```
  [LAMBDA (STREAM NOERROR COUNTP)                                                    ; Edited 25-Feb-91 16:35 by nm
```

;;; Returns a 16 bit XCCS code. Assuming the input character stream is encoded with EUC (Extended Unix Codes). Although EUC is independent of a
;;; particular language, the language implemented here is Japanese, thus this should be called as UJIS (Unixnized extended JIS code). JEIDA uses
;;; EUC as UJIS. Doesn't do EOL conversion -- \INCHAR or \INCCODE do that.

```
    (PROG ((CH1 (\PEEKBIN STREAM NOERROR))
           CH2)
          (COND
             [(NULL CH1)
              (COND
                 (COUNTP (RETURN (CL:VALUES NIL 0)))
                 (T (RETURN NIL)
             [(\EUC.KANJI.FIRST.BYTEP CH1)                                           ; Read next byte and compose a kanji character.
              (\BIN STREAM)                                                          ; Consume the first byte.
              [COND
                 ((NULL (SETQ CH2 (\PEEKBIN STREAM NOERROR)))
                  (\BACKFILEPTR STREAM)
                  (COND
```

```
                              (COUNTP (RETURN (CL:VALUES NIL 0)))
                              (T (RETURN NIL]
                     (\BACKFILEPTR STREAM)
                     (COND
                        (COUNTP (RETURN (CL:VALUES (\CONV.JIS.TO.XCCS (LOGAND CH1 127)
                                                                      (LOGAND CH2 127))
                                              0)))
                        (T (RETURN (\CONV.JIS.TO.XCCS (LOGAND CH1 127)
                                                      (LOGAND CH2 127]
               [(\EUC.HANKAKUP CH1)
                (\BIN STREAM)                                              ; Consume the SS2.
                [COND
                   ((NULL (SETQ CH2 (\PEEKBIN STREAM NOERROR)))
                    (\BACKFILEPTR STREAM)
                    (COND
                       (COUNTP (RETURN (CL:VALUES NIL 0)))
                       (T (RETURN NIL]
                (\BACKFILEPTR STREAM)
                (COND
                   (COUNTP (RETURN (CL:VALUES (\CONV.HANKAKU.KANA CH2)
                                              0)))
                   (T (RETURN (\CONV.HANKAKU.KANA CH2]
               [(\GAIJIP CH1)
                (\BIN STREAM)                                              ; Consume the SS3.
                [COND
                   ((NULL (SETQ CH1 (\PEEKBIN STREAM NOERROR)))
                    (\BACKFILEPTR STREAM)
                    (COND
                       (COUNTP (RETURN (CL:VALUES NIL 0)))
                       (T (RETURN NIL]
                (\BIN STREAM)                                              ; Consume the first byte in GAIJI.
                [COND
                   ((NULL (SETQ CH2 (\PEEKBIN STREAM NOERROR)))
                    (\BACKFILEPTR STREAM)
                    (\BACKFILEPTR STREAM)
                    (COND
                       (COUNTP (RETURN (CL:VALUES NIL 0)))
                       (T (RETURN NIL]
                (\BACKFILEPTR STREAM)
                (\BACKFILEPTR STREAM)
                (COND
                   (COUNTP (RETURN (CL:VALUES (\CONV.JIS.TO.XCCS (LOGAND CH1 127)
                                                                 (LOGAND CH2 127))
                                              0)))
                   (T (RETURN (\CONV.JIS.TO.XCCS (LOGAND CH1 127)
                                                 (LOGAND CH2 127]
               (T                                                          ; ASCII, C0, C1, SP or DEL
                  (COND
                     (COUNTP (RETURN (CL:VALUES CH1 0)))
                     (T (RETURN CH1])
```

(**BACKEUCCODE**

```
  [LAMBDA (STREAM COUNTP)                                                  ; Edited  6-Aug-2021 16:57 by rmk:
     (DECLARE (USEDFREE *BYTECOUNTER*))
     (CL:WHEN (\BACKFILEPTR STREAM)
        (IF (BITTEST (\PEEKBIN STREAM)
                (MASK.1'S 7 1))
            THEN                                                           ; C1, KAINJI, HANKAKU or GAIJI
               (IF (\BACKFILEPTR STREAM)
                   THEN (IF (\EUC.HANKAKUP (\PEEKBIN STREAM))
                            THEN (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -2))
                          ELSEIF (BITTEST (\PEEKBIN STREAM)
                                     (MASK.1'S 7 1))
                            THEN                                          ; KANJI or GAIJI
                               (IF (\BACKFILEPTR STREAM)
                                   THEN (IF (\GAIJIP (\PEEKBIN STREAM))
                                            THEN (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -3))
                                          ELSE              ; KANJI
                                               (\BIN STREAM)
                                               (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -2)))
                                 ELSEIF COUNTP
                                   THEN (SETQ *BYTECOUNTER* -2))
                          ELSE                                            ; C1
                               (\BIN STREAM)
                               (CL:WHEN COUNTP (SETQ *BYTECOUNTER* -1)))
                 ELSEIF COUNTP
                   THEN (SETQ *BYTECOUNTER* -1))
          ELSEIF COUNTP
            THEN (SETQ *BYTECOUNTER* -1)))])
)

(DEFINEQ
```

(**JISOUTCHARFN**

```
  [LAMBDA (OUTSTREAM CHARCODE)                                            ; Edited  8-Aug-2021 13:24 by rmk:
```

;;; Encoder for JIS format.

```
    (COND
        ((EQ CHARCODE (CHARCODE EOL))
         (FREPLACE (STREAM CHARPOSITION) OF STREAM WITH 0)
         (COND
            ((\KIMODEP OUTSTREAM NIL)
             (\OUTKO OUTSTREAM)
             (\CHNAGE.KI.MODE OUTSTREAM NIL NIL)))
         (\BOUTEOL OUTSTREAM))
        (T (CHANGE (FFETCH (STREAM CHARPOSITION) OF STREAM)
                   (IPLUS16 1 DATUM))
           (SETQ CHARCODE (\CONV.XCCS.TO.JIS OUTSTREAM CHARCODE))     ; \CONV.XCCS.TO.JIS converts ZENKAKUKATAKANA to
                                                                     ; HANKAKUKATAKANA if the stream desires it.
           (COND
              ((> CHARCODE 255)
               (COND
                  ((NOT (\KIMODEP OUTSTREAM NIL))
                   (\OUTKI OUTSTREAM)
                   (\CHNAGE.KI.MODE OUTSTREAM NIL T)))
               (\BOUT OUTSTREAM (\CHARSET CHARCODE))
               (\BOUT OUTSTREAM (\CHAR8CODE CHARCODE)))
              (T                                                     ; ASCII or HANKAKUKATAKANA
                 (COND
                    ((\KIMODEP OUTSTREAM NIL)
                     (\OUTKO OUTSTREAM)
                     (\CHNAGE.KI.MODE OUTSTREAM NIL NIL)))
                 (\BOUT OUTSTREAM CHARCODE])
```

(**SHIFTJISOUTCHARFN**
```
  [LAMBDA (OUTSTREAM CHARCODE)                                      ; Edited  8-Aug-2021 13:26 by rmk:
```

;;; Encoder for Shift-JIS format.

```
    (COND
        ((EQ CHARCODE (CHARCODE EOL))
         (FREPLACE (STREAM CHARPOSITION) OF STREAM WITH 0)
         (\BOUTEOL OUTSTREAM))
        (T (CHANGE (FFETCH (STREAM CHARPOSITION) OF STREAM)
                   (IPLUS16 1 DATUM))
           (SETQ CHARCODE (\CONV.XCCS.TO.JIS OUTSTREAM CHARCODE))     ; \CONV.XCCS.TO.JIS converts ZENKAKUKATAKANA to
                                                                     ; HANKAKUKATAKANA if the stream desires it.
           (COND
              [(> CHARCODE 255)
               (LET ((CH1 (FOLDLO CHARCODE 256))
                     (CH2 (LOGAND CHARCODE 255)))
                  (\CONV.JIS.TO.SJIS CH1 CH2)
                  (COND
                     ((AND (< CH1 256)
                           (< CH2 256))
                      (\BOUT OUTSTREAM CH1)
                      (\BOUT OUTSTREAM CH2]
              (T (\BOUT OUTSTREAM CHARCODE])
```

(**EUCOUTCHARFN**
```
  [LAMBDA (OUTSTREAM CHARCODE)                                      ; Edited  8-Aug-2021 13:27 by rmk:
```

;;; Encoder for EUC format.

```
    (COND
        ((EQ CHARCODE (CHARCODE EOL))
         (FREPLACE (STREAM CHARPOSITION) OF STREAM WITH 0)
         (\BOUTEOL OUTSTREAM))
        (T (CHANGE (FFETCH (STREAM CHARPOSITION) OF STREAM)
                   (IPLUS16 1 DATUM))
           (SETQ CHARCODE (\CONV.XCCS.TO.JIS OUTSTREAM CHARCODE))     ; \CONV.XCCS.TO.JIS converts ZENKAKUKATAKANA to
                                                                     ; HANKAKUKATAKANA if the stream desires it.
           (COND
              ((> CHARCODE 255)                                       ; KANJI or GAIJI
               (AND (NOT (\NOTGAIJIP CHARCODE))
                    (\BOUT OUTSTREAM 143))
               (\BOUT OUTSTREAM (LOGOR (\CHARSET CHARCODE)
                                       128))
               (\BOUT OUTSTREAM (LOGOR (\CHAR8CODE CHARCODE)
                                       128)))
              ((\HANKAKUP CHARCODE)
               (\BOUT OUTSTREAM 142)
               (\BOUT OUTSTREAM (LOGOR CHARCODE 128)))
              (T                                                     ; C0, C1, SP, DEL or G0
                 (\BOUT OUTSTREAM CHARCODE])
```

```
)
```

```
(DEFINEQ
```

(**CONVHANKAKU**
  [LAMBDA ARGS   ; Edited 8-Feb-91 13:42 by nm
    (PROG1 (STREAMPROP (ARG ARGS 1)
                 :HTOZP)
       (AND (> ARGS 1)
             (STREAMPROP (ARG ARGS 1)
                    :HTOZP
                    (ARG ARGS 2)))))])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS **CONV.XCCS.TO.JIS MACRO** (OPENLAMBDA (OUTSTREAM CC)

;;; Returns JIS code corresponding to XCCS charcode. Handle HANKAKU as well as ZENKAKU. If OUTSTREAM wants to convert ZENKAKUKANA to
;;; HANKAKUKANA, do so. Never returns two byte charcode for alpha-numeric character, they are all treated as single byte characode.

                                       (OR (COND
                                              ((\ASCIIP CC)
                                               CC)
                                              ((\NOT.EQUIVALENT.TO.JIS CC)
                                               (\DO.CONV.XCCS.TO.JIS CC))
                                              ((\CONV.HANKAKU.TO.ZENKAKUP OUTSTREAM)
                                                                ; ZENKAKUKANA comes here, because their charcodes are
                                                                ; equiavalent to JIS.
                                               (\CONV.ZENKAKU.KANA CC))
                                              (T CC))
                                          CC)))

(PUTPROPS **DO.CONV.XCCS.TO.JIS MACRO** ((CC)
                                          (GETHASH CC *XCCS-TO-JIS-CONV-TABLE*)))

(PUTPROPS **ASCIIP MACRO** (OPENLAMBDA (CC)
                              (AND (EQ (FOLDLO CC 256)
                                       0)
                                   (< (LOGAND CC 255)
                                      128))))

(PUTPROPS **NOT.EQUIVALENT.TO.JIS MACRO** (OPENLAMBDA (CC)
                                             (OR (EQ (FOLDLO CC 256)
                                                     0)
                                                 (EQ (FOLDLO CC 256)
                                                     33)
                                                 (EQ (FOLDLO CC 256)
                                                     34)
                                                 (EQ (FOLDLO CC 256)
                                                     38)
                                                 (EQ (FOLDLO CC 256)
                                                     40)
                                                 (EQ (FOLDLO CC 256)
                                                     239)
                                                 (EQ (FOLDLO CC 256)
                                                     241))))

(PUTPROPS **CONV.HANKAKU.TO.ZENKAKUP MACRO** ((OUTSTREAM)
                                               (STREAMPROP OUTSTREAM :HTOZP)))

(PUTPROPS **CONV.ZENKAKU.KANA MACRO** ((CHAR)
                                        (GETHASH CHAR *ZENKAKU-TO-HANKAKU-CONV-TABLE*)))
)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS **EXTRACT.NO.FONT.CODE MACRO** ((JISCODE)
                                          (CDR (ASSOC JISCODE *JIS-TO-XCCS-CONV-NO-FONT-TABLE*))))

(PUTPROPS **EXTARACT.CONV.TABLE MACRO** ((KU)
                                         (CDR (ASSOC KU *JIS-TO-XCCS-CONV-TABLE-LIST*))))

(PUTPROPS **NOT.EQUIVALENT.TO.XCCS MACRO** ((KU)

;;; The JIS codes which are not equiavelent to XCCS reside in 1, 2, 3, 6, 8 and 84 KU. Although from 84-5 to 94-94 inclusive are not defined in JIS, that
;;; is they are GAIJI, they are also handled here.

                                            (OR (EQ KU 33)
                                                (EQ KU 34)
                                                (EQ KU 35)
                                                (EQ KU 38)
                                                (EQ KU 40)
                                                (EQ KU 116)

```
                                                  (EQ KU 117))))

(PUTPROPS \EXTRACT.SET MACRO ((TEN TABLE)
                             (ELT TABLE (IDIFFERENCE (UNFOLD (IDIFFERENCE TEN 32)
                                                            2)
                                        1))))

(PUTPROPS \EXTRACT.CODE MACRO ((TEN TABLE)
                              (ELT TABLE (UNFOLD (IDIFFERENCE TEN 32)
                                                 2))))

(PUTPROPS \CHNAGE.KI.MODE MACRO [OPENLAMBDA (ST INPUTFLG ENTERP)
```

;;; INPUTFLG is true if \CHNAGE.KI.MODE is called in the context in which ST is an input stream.

```
                                     (COND
                                        [INPUTFLG (COND
                                                     (ENTERP (freplace (STREAM IN.KANJIIN)
                                                                 of (\DTEST ST 'STREAM) with T))
                                                     (T (freplace (STREAM IN.KANJIIN) of (\DTEST ST 'STREAM)
                                                             with NIL]
                                           (T (COND
                                                 (ENTERP (freplace (STREAM OUT.KANJIIN) of (\DTEST ST 'STREAM)
                                                              with T))
                                                 (T (freplace (STREAM OUT.KANJIIN) of (\DTEST ST 'STREAM) with NIL])

(PUTPROPS \KIMODEP MACRO [OPENLAMBDA (ST INPUTFLG)
```

;;; INPUTFLG is true if \KIMODEP is called in the context in which ST is an input stream.

```
                                     (COND
                                        [INPUTFLG (ffetch (STREAM IN.KANJIIN) of (\DTEST ST 'STREAM]
                                        (T (ffetch (STREAM OUT.KANJIIN) of (\DTEST ST 'STREAM])

(PUTPROPS \HANKAKUP MACRO ((CHAR)
                          (< 160 CHAR 224)))

(PUTPROPS \KANJIP MACRO ((CHAR)
                        (< 12158 CHAR 29733)))

(PUTPROPS \NOTGAIJIP MACRO ((CHAR)
                           (OR (< 8480 CHAR 10305)
                               (< 12158 CHAR 29733))))

(PUTPROPS \INVALID.TENP MACRO (OPENLAMBDA (TEN)
                              (OR (< TEN 33)
                                  (< 126 TEN))))

(PUTPROPS \CONV.HANKAKU.KANA MACRO ((CHAR)
                                   (GETHASH CHAR *HANKAKU-TO-ZENKAKU-CONV-TABLE*)))

(PUTPROPS \OUTKI MACRO ((STREAM)
                       (\BOUT OUTSTREAM (CHARCODE ESC))
                       (\BOUT OUTSTREAM (CHARCODE $))
                       (\BOUT OUTSTREAM (CHARCODE B))))

(PUTPROPS \OUTKO MACRO ((STREAM)
                       (\BOUT OUTSTREAM (CHARCODE ESC))
                       (\BOUT OUTSTREAM (CHARCODE %())
                       (\BOUT OUTSTREAM (CHARCODE J))))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \CONV.SJIS.TO.JIS MACRO [OPENLAMBDA (HI LO)
```

;;; Convert Shift-JIS to JIS. The variable named CH1 and CH2 are set to the converted hight 8 bit and low 8bit of JIS code respectively.

```
                                     [SETQ CH1 (IDIFFERENCE HI (COND
                                                                  ((> HI 159)
                                                                   177)
                                                                  (T 113]
                                     (SETQ CH1 (IPLUS (UNFOLD CH1 2)
                                                      1))
                                     (SETQ CH2 (COND
                                                  [(> LO 158)
                                                   (PROG1 (IDIFFERENCE LO 126)
                                                       (SETQ CH1 (IPLUS CH1 1)))]
                                                  (T (IDIFFERENCE LO (COND
                                                                        ((> LO 126)
                                                                         (IPLUS 31 1))
                                                                        (T 31])

(PUTPROPS \CONV.JIS.TO.SJIS MACRO [OPENLAMBDA (HI LO)
```

;;; Convert JIS to Shift-JIS. The variable named CH1 and CH2 are set to the converted hight 8 bit and low 8bit of Shift-JIS code respectively.

```
                                          [SETQ CH2 (COND
                                                      ((ODDP HI)
                                                       (SETQ CH2 (IPLUS LO 31)))
                                                      (COND
                                                         ((>= CH2 127)
                                                          (IPLUS CH2 1))
                                                         (T CH2)))
                                                      (T (IPLUS LO 126]
                                          (SETQ CH1 (IPLUS (FOLDLO (IDIFFERENCE HI 33)
                                                                   2)
                                                           129))
                                          (AND (> CH1 159)
                                               (SETQ CH1 (IPLUS CH1 64])

(PUTPROPS \SJIS.KANJI.FIRST.BYTEP MACRO (OPENLAMBDA (CHAR)
                                           (OR (< 127 CHAR 160)
                                               (< 223 CHAR 256))))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \EUC.KANJI.FIRST.BYTEP MACRO ((CHAR)
                                        (< 160 CHAR 255)))

(PUTPROPS \GAIJIP MACRO ((CHAR)
                         (EQ CHAR 143)))

(PUTPROPS \EUC.HANKAKUP MACRO ((CHAR)
                               (EQ CHAR 142)))
)
)

(DEFINEQ
```

(**CREATE.JIS.EXTERNALFORMAT**
```
  [LAMBDA NIL                                               ; Edited  5-Aug-2021 22:40 by rmk:
```

;;; Create an instance of EXTERNALFORMAT datatype and install it with :JIS as its name.

```
    (MAKE-EXTERNALFORMAT :JIS (FUNCTION \JISIN)
          (FUNCTION \JISPEEK)
          (FUNCTION \BACKJISCCODE)
          (FUNCTION \JISOUTCHARFN])
```

(**CREATE.SHIFTJIS.EXTERNALFORMAT**
```
  [LAMBDA NIL                                               ; Edited  5-Aug-2021 22:41 by rmk:
```

;;; Create two instances of EXTERNALFORMAT datatype and install them with :W-MS and :MS as their names respectively. :MS have to change the
;;; end of line convention to CRLF.

```
    (MAKE-EXTERNALFORMAT :W-MS (FUNCTION \SHIFTJISIN)
          (FUNCTION \SHIFTJISPEEK)
          (FUNCTION \BACKSHIFTJISCCODE)
          (FUNCTION \SHIFTJISOUTCHARFN))
    (MAKE-EXTERNALFORMAT :MS (FUNCTION \SHIFTJISIN)
          (FUNCTION \SHIFTJISPEEK)
          (FUNCTION \BACKSHIFTJISCCODE)
          (FUNCTION \SHIFTJISOUTCHARFN)
          NIL
          'CRLF])
```

(**CREATE.EUC.EXTERNALFORMAT**
```
  [LAMBDA NIL                                               ; Edited  5-Aug-2021 22:40 by rmk:
```

;;; Create an instance of EXTERNALFORMAT datatype and install it with :EUC as its name.

```
    (MAKE-EXTERNALFORMAT :EUC (FUNCTION \EUCIN)
          (FUNCTION \EUCPEEK)
          (FUNCTION \BACKEUCCODE)
          (FUNCTION \EUCOUTCHARFN])

)

(DECLARE%: DONTEVAL@LOAD DOCOPY
```

(**CREATE.JIS.EXTERNALFORMAT**)

(**CREATE.SHIFTJIS.EXTERNALFORMAT**)

(**CREATE.EUC.EXTERNALFORMAT**)
```
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(OR (GETP 'EXPORTS.ALL 'FILE)
```

```
        (PRINT "NOTE: JAPANESE requires EXPORTS.ALL for compilation" T))
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA CONVHANKAKU)
)
```

FUNCTION INDEX

MACRO INDEX

VARIABLE INDEX