(Author: Ron Kaplan)

The XCCS files contain mappings from the Xerox Character Code Standard into Unicode 3.0. standard codes. Originally, the mapping files were constructed heuristically for XCCS Version XC1-3-3-0 (1987) from incomplete and possibly errorful sources. But those files corresponded well enough to the (incomplete) fonts in the Medley system. The current files were produced more systematically by Peter Craven, and correspond to XCCS Version 2.0, 1990. These files generally provide a superset of the mappings of the original files, and some incorrect mappings have been fixed. The newer files also contain the character names for almost all the characters. The Unicode directory contains a pdf of the 1990 Xerox standard.

Each file may contain the mappings for one or more XCCS character sets. By convention, the name of the file indicates the character set mappings it contains. A file with a single mapping has a name of the form XCCS-<csnumber>=<csname>.TXT, where csnumber is the octal number of the character set and csname is a cover term for its mappings. For example, XCCS-341=HEBREW.TXT contains the mappings for Hebrew.

If a file contains several character sets, its name specifies just the numbers of those sets. For example, XCCS-0,41-50,340-344,356-361.TXT contains mappings for character set 0, 41 through 50, 340 through 344, and 356 through 361 (basically, all the non JIS character sets).

The format of each file conforms to the format of the other Unicode-supplied mapping files:
    Three white-space (tab or spaces) separated columns:
        Column 1 is the XCCS code (as hex 0xXXXX)
        Column 2 is the corresponding Unicode (as hex 0xXXXX)
        Column 3 (after #) is a comment column.
            For convenience, it contains the Unicode character itself, and then the
            Unicode names as available.

In some of the files Unicode FFFF is used for the piece-meal (as opposed to systematic) undefined XCCS codes (Column 3 is UNDEFINED), in other files undefined mappings for particular XCCS codes are simply omitted. Presumably undefined XCCS codes will never appear in XCCS files.

Unicode FFFE is used for defined XCCS codes whose Unicode mapping has not been determined (Column 3 is MISSING). These may be rare, but until/unless these are filled in, XCCS documents contain them them will not be properly represented in Unicode. Thus, this value flags the need for additional Unicode sleuthing.

Note: A very few of the XCCS codes map to Unicodes that are outside the 16 bit plane. Those codes cannot be interpreted inside Medley, and those lines in the mapping files have therefore been commented out. Those lines can be identified in the future because their UTF-8 character-bytes have also been replaced by XXXX.

Like the other Unicode mapping files, this file can be read by common Unicode routines. Also, it is encoded in UTF-8, so that the Unicode characters are properly displayed in Column 3 and can be edited by standard Unicode-enabled editors (e.g. Mac Textedit).

These files and the mapping files in sister directories can also be read by the function READ-UNICODE-MAPPING in the UNICODE Medley library package, and they can be written by WRITE-UNICODE-MAPPING.

The entries are in XCCS order and grouped by character sets. In front of each character set, for convenience, there is a #-comment line with the octal XCCS character set and the character-set name (e.g. # "341" HEBREW).

Note that a given XCCS code might map to codes in several different Unicode positions, since there are repetitions in the Unicode standard.

Any comments or problems, contact <ron.kaplan@post.harvard.edu>

The original mappings were derived from the file XCCStoUni whose provenance is unknown, and there is no specification for its structure. It appears to be a sequence of 2-byte hex Unicode characters with all the Unicode characters in a given character set laid out in ascending XCCS code order.

It seems to have entries only for 188 characters per character set, with no 2-byte cells for the undefined regions of the two 128 code panels of each XCCS character set. So code 127 is skipped at each panel boundary and the running XCCS code is then bumped by 33. The hexcode at file position 0 is for octal 41 (exclamation mark); the space 40 isn't represented.

Within that, Unicode FFFD (the unicode missing-character slug) is used for XCCS codes whose Unicode equivalent is not specified, and it seems that FFFF is used when whole panels are missing (the higher order panel for most of the Japanese).

Finally, no cells are allocated for the unused/reserved character sets (1 through octal 40), so that the Unicode after octal 376 is for 41,41. But the order of character sets is a little jumbled, so that the JIS character sets (60 through 163), for example, start at 75 (octal) in the file sequence--some higher number character sets appear earlier in the file than they should.

The JIS character sets seem to be complete and accurate. There are sporadic missing codes and errors in some of the other sets that required hand correction.

Here is more information about that file, particularly about the unknown blocks at the end of the file, as provided by Peter Craven

It contains blocks of two times 94 characters, mapping to the left and right side of Character Sets. The blocks occur as follows in the file:

```
- character set 000 (block 0)
- character set 041 (block 1)
- character set 042 (block 2)
...
- character set 050 (block 8)
- character set 052 (block 9) ;; extended cyrillic
- character set 057 (block 10) ;; vertically written japanese symbols
- character set 164 (block 11) ;; miscellaneous japanese symbols
- character set 165 (block 12) ;; jis extra
- character set 166 (block 13) ;; symbols 4
- character set 340 (block 14) ;; arabic
- character set 341 (block 15) ;; hebrew
- character set 342 (block 16) ;; ipa
- character set 343 (block 17) ;; korean
- character set 344 (block 18) ;; armenian/georgian
- character set 345 (block 19) ;; devanagari
- character set 346 (block 20) ;; bengali
- character set 347 (block 21) ;; gurmukhi
- character set 350 (block 22) ;; thai
- character set 353 (block 23) ;; general and technical symbols 3
- character set 354 (block 24) ;; extended itc dingbats 2 and general symbols
- character set 355 (block 25) ;; itc dingbats 1
- character set 356 (block 26) ;; general and technical symbols 2
- character set 357 (block 27) ;; general and technical symbols 1
- character set 365 (block 28) ;; initial, medial, and final arabic characters
- character set 060 (block 29) ;; CJK
...
- character set 163 (block 96) ;; CJK
```

Blocks 97 to 148 (which all contain both a left and a right side) are Chinese characters (to my limited knowledge some of these are definitely not Japanese), but I have no idea which standard they follow.

94 * 2 * 2 (two bytes for each left and right side) * 149 (number of blocks) = 56024, which matches the total file size exactly.

And a little more, from John Cowan, about the unknown character sets at the end:

The order of the first few characters matches with this list of 3000 most frequent Hanzi[fn:1] almost exactly, when looking at the Traditional Forms. So this is probably an attempt to encode Chinese in order of frequency. This seems to encode 94 * 2 * 52 blocks = 9776 characters.

See https://onedrive.live.com/?authkey=%21AFZ83zRXy1tjL%5Fo&cid=A0A919DEAF7F6BC7&id=A0A919DEAF7F6BC7%21228&parId=root&o=OneUp