UNICODE

By Ron Kaplan

This document was last edited in March 2024.

The UNICODE library package defines external file formats that enable Medley to read and write files where 16 bit character codes are represented as UTF-8 byte sequences or UTF-16 byte-pairs. It also provides for character codes to be converted (on reading) from Unicode codes to equivalent codes in the Medley-internal Xerox Character Code Standard (XCCS) and (on writing) from XCCS codes to equivalent Unicode codes.

Unicode external formats

Four external formats are defined when the package is loaded:

:UTF-8 codes are represented as UTF-8 byte sequences and XCCS/Unicode character conversion takes place.

:UTF-16BE codes are represented as 2-byte pairs, with the high order byte appearing first in the file, and characters are converted.

:UTF-16LE codes are represented as 2-byte pairs, with the low order byte appearing first in the file, and characters are converted.

The two other external formats translate byte sequences into codes, but do not translate the codes. These allow Medley to see and process characters in their native encoding.

:UTF-8-RAW codes are represented as UTF-8 byte sequences, but character conversion does not take place.

:UTF-16BE-RAW codes are represented as big-ending 2-byte pairs but there is no conversion.

:UTF-16LE-RAW codes are represented as little-ending 2-byte pairs but there is no conversion.

These formats all define the end-of-line convention (mostly for writing) for the external files according to the variable EXTERNALEOL (LF, CR, CRLF), initially set to LF.

The external format can be specified as a parameter when a stream is opened:

```
(OPENSTREAM 'foo.txt 'INPUT 'OLD '((EXTERNALFORMAT :UTF-8)))
```

```
(CL:OPEN 'foo.txt :DIRECTION :INPUT :EXTERNAL-FORMAT :UTF-8)
```

The opening parameters may be overridden if READBOM is invoked by the calling function (e.g. Tedit) and it detects a byte-order-mark at the beginning of the file:

(READBOM STREAM) [Function]
(WRITEBOM STREAM FORMAT) [Function]

READBOM returns one of :UTF-8, :UTF-16BE or :UTF-16LE if a BOM is present, otherwise NIL, WRITEBOM writes BOM bytes only if given a Unicode FORMAT.

The function STREAMPROP obtains or changes the external format of an open stream:

```
(STREAMPROP stream 'EXTERNALFORMAT) -> :XCCS
```

```
(STREAMPROP stream 'EXTERNALFORMAT :UTF-8) -> :XCCS
```

In the latter case, the stream's format is changed to :UTF-8 and the previous value is returned, in this example it is Medley's historical default format :XCCS.

Entries can be placed on the variable *DEFAULT-EXTERNALFORMATS* to change the external format that is set by default when a file is opened on a particular device. Loading UNICODE executes

```
(PUSH *DEFAULT-EXTERNALFORMATS* '(UNIX :UTF-8))
```

so that all (non-BOM) files opened by OPENSTREAM, CL:OPEN, etc. on the UNIX file device will be initialized with :UTF-8. Note that the UNIX and DSK file devices reference the same files (although some caution is needed because {UNIX} does not simulate Medley versioning), but the device name in a file name ({UNIX}/Users/... vs. {DSK}/Users/...) selects the particular device. The default setting above applies only to files specified with {UNIX}; a separate default entry for DSK must be established to change its default from :XCCS.

The user can also specify the external format on a per-stream basis by putting a function on the list STREAM-AFTER-OPEN-FNS. After OPENSTREAM opens a stream and just before it is returned to the calling function, the functions on that list are applied in order to arguments STREAM, ACCESS, PARAMETERS. They can examine and/or change the properties of the stream, in particular, by calling STREAMPROP to change the external format from its device default.

Translating between Unicode and XCCS character codes

The external formats use the primitive macro UNICODE.TRANSLATE to map between XCCS and Unicode codes.

(UNICODE.TRANSLATE CODE TRANSLATION-TABLE) [Macro]

TRANSLATION-TABLE is a XCCS/Unicode mapping table as defined by the code-mapping files for a particular collection of XCCS character sets. These are located in a subdirectory of the top-level MEDLEYDIR directory, as determined by the value of UNICODEDIRECTORIES.

UNICODEDIRECTORIES [Variable]

This is initialized to (>Unicode>Xerox> >Unicode>Xerox>JIS>).

The functions described below construct the translation tables that UNICODE.TRANSLATE makes use of. The tables for an initial default collection of character sets are created during the system-building process. The translation tables for other character sets can be installed in anticipation of work on files that contain those characters. However, UNICODE.TRANSLATE will also install new character sets on demand, the first time an unmapped character is encountered.

The mapping files have conventional names of the form XCCS-[charsetnum]=[charsetname].TXT, for example, XCCS-0=LATIN.TXT, XCCS-357=SYMBOLS4.TXT. The translations used by the external formats are read from these files by the function

(READ-UNICODE-MAPPING FILESPEC NOPRINT NOERROR) [Function]

where FILESPEC can be a list of files, charset octal strings ("0" "357"), or XCCS charset names (LATIN EXTENDED-LATIN GREEK), or XCCS character codes. A character code is taken to indicate the XCCS character set that it belongs to. Reading will be silent if NOPRINT, and the process will not abort if an error occurs and NOERROR. The value is a flat list of the mappings for all the indicated character sets, with elements of the form (XCCC-code Unicode-code).

READ-UNICODE-MAPPING uses READ-UNICODE-MAPPING-FILENAMES to interpret the FILESPEC.

(READ-UNICODE-MAPPING-FILENAMES FILESPEC) [Function]

converts the list of mapping-file specifications into a list of corresponding files in any of the directories in UNICODEDIRECTORIES. If a file specification is the name of a subdirectory it will expand to the names of all of the mapping files in that subdirectory. Thus JIS will result in a list of all of the JIS>XCCS-*=JIS.TXT files.

The mapping from character-set names to octal character-set numbers is provided by the entries on XCCS-CHARSETS.

XCCS-CHARSETS [Variable]

This is an alist whose entries are of the form (NAME OCTAL-CHARSET), for example (GREEK "46"). An entry can also denote a collection of character-sets to be installed all together. Thus the set of character sets that are installed when UNICODE is loaded is specified by the entry

```
(DEFAULT LATIN ACCENTED-LATIN1 EXTENDED-LATIN SYMBOLS1 SYMBOLS2 FORMS
JAPANESE-SYMBOLS1 JAPANESE-SYMBOLS2).
```

Similarly, the collection of character sets needed to represent the Unicode mapping for Japanese are specified by the entry

```
(JAPANESE HIRAGANA KATAKANA JIS).
```

JIS in this entry denotes all of the mapping tables in the JIS subdirectory.

The internal translation tables used by the external formats are constructed from a list of correspondence pairs by the function

(MAKE-UNICODE-TRANSLATION-TABLES MAPPING REINSTALL) [Function]

MAPPING is a list of correspondence pairs as provided by READ-UNICODE-MAPPING or a FILESPEC that can be given to that function. This creates and returns a list of two arrays (XCCS-to-Unicode Unicode-to-XCCS) containing the relevant translation information organized for rapid access. It may also install the mapping arrays in two global variables:

XCCSTOUNICODE [Global variable]
UNICODETOXCCS [Global variable]

If *XCCSTOUNICODE* is NIL (initially) or if REINSTALL, the variables *XCCSTOUNICODE* and *UNICODETOXCCS* are set to the respective arrays. The top-level values of these variables are used by the external-format functions to perform translations in either direction.

The function MERGE-UNICODE-TRANSLATION-TABLES can be used to merge additional character-set translations into existing target tables.

(MERGE-UNICODE-TRANSLATION-TABLES ADDITION TABLE
INVERSETABLE) [Function]

ADDITION is a mapping-array pair, a list of mapping pairs, or a FILESPEC for a mapping to be constructed. TABLE is the mapping table (*XCCSTOUNICODE* or *UNICODETOXCCS*) for the domain of the addition, INVERSETABLE is the mapping table for its range.

The character-set mapping files are quite verbose, with textual information that makes it easy to read and edit but is not necessary for run-time decoding. The files also are organized so that translations for given XCCS characters and character sets are easily retrieved but it is less efficient to find the XCCS character set that holds the XCCS code for a given Unicode code. There are two additional files in the Unicode directory, UNICODE-MAPPINGS.TXT and INVERTED-UNICODE-MAPPINGS.TXT, that cache for all character sets the forward and backward information needed for run-time decoding. The macro UNICODE.TRANSLATE uses these files to create internal mapping tables for character-correspondences that have not yet been isntalled.

These files are produced by the functions

(ALL-UNICODE-MAPPINGS FILE) [Function]
(INVERT-ALL-UNICODE-MAPPINGS FILE) [Function]

If FILE is NIL, these return the respective list structures. If FILE is T, the mapping files named above are created. Other non-NIL values are taken as the names of files that the lists should be written to.

There is also a function for writing verbose textual mapping files given a list of mapping pairs:

(WRITE-UNICODE-MAPPING MAPPING INCLUDEDCHARSETS FILE) [Function]

produces one or more textual mapping files for the mapping-pairs in MAPPING. If the optional FILE argument is provided, then a single file with that name will be produced and contain all the mappings for all the character sets in MAPPING. If FILE and INCLUDEDCHARSETS are not provided, then all of the mappings will again go to a single file with a composite name XCCS-csn1,csn2,csn3.TXT. Each cs may be a single charset number, or a range of adjacent charset numbers. For example, if the mappings contain entries for characters in charset LATIN, SYMBOLS1, SYMBOLS2, and EXTENDED-LATIN, the file name will be XCCS-0,41-43.TXT.

If INCLUDEDCHARSETS is provided, it specifies possibly a subset of the mappings in MAPPING for which files should be produced. This provides an implicit subsetting capability.

Finally, if FILE is not provided and INCLUDEDCHARSETS is T, then a separate file will be produced for each of the character sets, essentially a way of splitting a collection of character-set mappings into separate canonically named files (e.g. XCCS-357=SYMBOLS1.TXT).

Additional UTF-8 and Unicode functions

The following utilities are provided for lower-level manipulation of codes and strings.

(UTF8.BINCODE STREAM RAW) [Function]

Reads bytes of a UTF-8 code starting at the current postion of STREAM, returning the code represented by those bytes. The code is translated to its XCCS equivalent unless RAW. Unlike the INCCODEFN of the external format, this does not do any special EOL interpretation. STREAM is positioned after the last byte read.

(UTF8.VALIDATE STREAM BYTE) [Function]

Reads bytes starting at the current postion of STREAM until reaching the end of a valid UTF-8 encoding. If BYTE is provided, it is interpreted as the first (already-read) byte of the sequence.

Otherwise the first byte is read. If the byte sequence is valid, returns the length of the encoding, otherwise NIL. Either way, `STREAM` is position after the last byte read.

`(NUTF8-BYTE1-BYTES BYTE1)` [Function]

Returns the number of bytes in a UTF-8 code representation whose first byte is BYTE1.

`(NUTF8-CODE-BYTES CODE)` [Function]

Returns the number of bytes in the UTF-8 representation of `CODE`.

`(NUTF8-STRING-BYTES STRING RAW)` [Function]

Returns the number of bytes in the UTF-8 representation of `STRING`, translating XCCS to Unicode unless `RAW`.

`(XTOUCODE XCCSCODE)` [Function]

Returns the Unicode code corresponding to `XCCSCODE`.

`(UTOXCODE UNICODE)` [Function]

Returns the XCCS code corresponding `UNICODE`.

`(XTOUSTRING XCCSSTRING RAW)` [Function]

Returns the string of bytes in the UTF-8 representation of the characters in XCCSSTRING (= the bytes in its UTF-8 file encoding).

`(HEXSTRING N WIDTH)` [Function]

Returns the hex string for N, padded to `WIDTH.`