READ-BDF

By: Matt Heffron (heffron@alumni.caltech.edu)

This document last edited on April 26, 2025.

INTRODUCTION

This module defines functions to read BDF format font files, and then to write DISPLAYFONT files containing the glyphs from the source BDF file. In normal use, these DISPLAYFONT files will have remapped the glyphs from their Unicode encoding to Medley's XCCS encoding. All newly created symbols in this module are in the **BDF** package.

FUNCTIONS

(BDF:READ-BDF *PATH* &OPTIONAL *VERBOSE*) [Function]

BDF:READ-BDF reads and parses the BDF file specified by *PATH*. (It is required and does not default the extension of the filename.) This returns a BDF::BDF-FONT structure containg ***all*** of the glyphs contained in the file. If *VERBOSE* is non-NIL, then at, its completion, BDF:READ-BDF will print the full internal font name, and the values of the Family, Size, Weight, Slant, and Expansion as determined from the parsed BDF file. (Collectively, Weight, Slant, and Expansion comprise the font FACE. See the IRM 27.12, **Fonts**, for further description.)

(BDF:WRITE-BDF-TO-DISPLAYFONT-FILES *BDFONT DEST-DIR* &KEY *FAMILY SIZE FACE ROTATION DEVICE CHAR-SETS MAP-UNKNOWN-TO-PRIVATE WRITE-UNMAPPED RAW-UNICODE-MAPPING*) [Function]

BDF:WRITE-BDF-TO-DISPLAYFONT-FILES writes DISPLAYFONT files from *BDFONT* for each XCCS character set which contains glyphs mapped into it. *BDFONT* must be of a BDF::BDF-FONT structure. The files will be written into the corresponding sub-directory of *DEST-DIR* by character set number. *BDFONT* and *DEST-DIR* **must** be provided.

FAMILY, *SIZE*, *FACE*, *ROTATION*, and *DEVICE* correspond to the same arguments to IL:FONTCREATE. BDF:WRITE-BDF-TO-DISPLAYFONT-FILES will attempt to get default values from the information in the BDF::BDF-FONT. (See the description of *VERBOSE* output of BDF:READ-BDF, *above*.) *FAMILY* may be a list of values as for IL:FONTCREATE, in which case *SIZE FACE ROTATION* and *DEVICE* are taken from the *FAMILY* list. *FAMILY* may also be an IL:FONTDESCRIPTOR, again as for IL:FONTCREATE, in this case the values will be used if no values were specified or found in the BDF::BDF-FONT.

CHAR-SETS specifies which character sets, if present, should have the corresponding DISPLAYFONT files written. It is either a single character set number (integer from 0 to 255), a list of character set numbers, T meaning all character sets, or NIL meaning only character set 0. The default is T.

MAP-UNKNOWN-TO-PRIVATE selects how glyphs (character codes) which are present in the BDF file but do not map into XCCS should be handled. If non-NIL, then those will be mapped into the private range of the XCCS encoding space, and the corresponding DISPLAYFONT files will be written. If *MAP-UNKNOWN-TO-PRIVATE* is NIL (the default), then, if *WRITE-UNMAPPED* is non-NIL, they will be written, with their BDF (Unicode) encoding, into a parallel set of DISPLAYFONT files with "-UNMAPPED" appended to the FAMILY part of the file name.

RAW-UNICODE-MAPPING, if non-NIL, overrides the mapping of character codes from the BDF file (Unicode) encoding to XCCS encoding, and just writes the DISPLAYFONT files according to the BDF file encoding for the glyphs. It will prefix the FAMILY part of the file name with "RAW-". This defaults to NIL.

BDF:WRITE-BDF-TO-DISPLAYFONT-FILES returns five values (not a *list* of five values). The IL:FONTDESCRIPTOR of the mapped character sets, the list of character sets with DISPLAYFONT files actually written, the IL:FONTDESCRIPTOR of the unmapped character sets, the list of the unmapped character sets with DISPLAYFONT files actually written, the list of BDF::GLYPH instances which could not be mapped. Specifically, those glyphs which have no mapping into XCCS and with source encoding either > 0xFFFF or with a low byte = 0xFF. These glyphs are never written to DISPLAYFONT files.

NOTES

- The names of the two *entry point* functions, READ-BDF and WRITE-BDF-TO-DISPLAYFONT-FILES, are the only symbols *exported* from the BDF package, all others are *internal* to that package.
- Using a FAMILY name that contains any digits, either explicitly or from the information in the BDF::BDF-FONT, is **strongly discouraged**. This is likely to result in fonts which cannot be located by the standard font search process (e.g., TEdit), although they can be created *explicitly* using IL:FONTCREATE.
- It is *recommended* that *DEST-DIR* provided to BDF:WRITE-BDF-TO-DISPLAYFONT-FILES be a different location than those on the *standard value* of IL:DISPLAYFONTDIRECTORIES. This will protect your files from possible loss when newer releases of Medley are installed. You may (in your personal INIT file) add your *DEST-DIR* location to the IL:DISPLAYFONTDIRECTORIES list to enable the normal font finding process to locate your fonts.
- For BDF files containing a large number of glyphs, BDF:READ-BDF can take quite a while and use a lot of memory. For example, READ-BDF of gnu unifont-16.0.02.bdf with 57086 glyphs, took 72 seconds, and allocated 1.3 million FIXP and 3.3 million ONED-ARRAY.)