CEDILLA

¸

A best-effort text printer

Juliusz Chroboczek
<jch@pps.jussieu.fr>

Cedilla is a simple text printer that uses Unicode internally.

Using Unicode means that the set of characters that can appear in the input is very large, and the user may very well have no font available that contains glyphs for the characters that he wants to print. Cedilla attempts to at least partially solve this problem using a number of techniques:

1. Cedilla can use an arbitrary number of downloadable fonts; for any given print job, only the necessary fonts will be downloaded.

2. Cedilla will use its own built-in font which contains a number of useful glyphs that are missing from standard fonts.

3. Cedilla will modify existing glyphs in order to e.g. remove dots or add bars.

4. Cedilla will attempt to build composite glyphs (e.g. for accented characters) on the fly.

5. Cedilla will use fallbacks for characters that are not supported by the available fonts.

SOURCES OF GLYPHS

Cedilla will attempt to use glyphs from all of the following sources.

1. Glyphs present in an available font.

This is the common case, and covers all simple characters but also, depending on the used font, a number of composites.

Example: Être ou ne pas être ?

2. Built-in glyphs.

Cedilla has a built-in font parts of which will be downloaded if needed.

Example: €.

3. Modified glyphs

When no suitable glyph is available, Cedilla will sometimes attempt to retouch an available glyph. The main application is to produce dotless glyphs for further composition; another use is to provide barred or doubled glyphs.

Example: Međunarodna željeznička unija.

This mechanism is also used to provide circled and more generally enclosed glyphs; this is not expected to be useful in practice.

4. Transformed glyphs

Cedilla is able to linearly transform (rotate, translate, mirror) a glyph in a font. The main application is to provide a turned comma for further compositing.

5. Glyphs composed based on data accompanying the font.

The Adobe Font Metrics (AFM) format can include positioning information for composites, and Cedilla will be glad to use such data. However, as few fonts come with extra information in this form, this technique is seldom useful.

6. Glyphs composed out of components present in a single font.

If a glyph is missing from a font, but all the components needed to construct it are present, Cedilla will build a composite glyph out of those. While the positioning of the diacritical marks is approximate, the algorithm used appears to be satisfactory with many standard fonts.

Example: Czy pamiętasz jak ze mną tańczyłaś Walca ?

When building such composites, Cedilla will, whenever necessary, replace dotted letters with their dotless variants or modify the base glyph to remove the dot (see Section 2 above).

Example: Tiuj ĉi arĥaismoj neniam estos elĵetitaj.

Of course, Cedilla is not limited to using glyphs that are present in Unicode in precomposed form. For example, the second letter of the third word in the sentence below is a Latin small letter e followed with 'Combining Vertical Line Below.'

Example: Mo lè je̩ dígí, kò ní pa mí lára.

Multiple combining characters may apply to a single base character, and will usually be stacked in the order mandated by Unicode.

Example: í̇ (i◌́ ◌̇) is different from i̇́ (i◌̇ ◌́) which is different from í (i◌́).

In some cases, however, they will be positioned in a specific, culturally acceptable manner.

Example: Hành lang hôi mùi cải luộc chiếu nát.

7. Glyphs composed out of components taken from different fonts.

When the font containing a selected base glyph doesn't contain suitable diacritical marks, Cedilla will search for a diacritical mark in the other fonts available. This can be useful in some cases, and will in particular enable printing almost legible Greek when no decent Greek font is available:

Example:
Οὐχὶ ταὐτὰ παρίσταταί μοι γιγνώσκειν, ὦ ἄνδρες ᾿Αθηναῖοι,
ὅταν τ᾿εἰς τὰ πράγματα ἀποβλέψω καὶ ὅταν πρὸς τοὺς
λόγους οὓς ἀκούω·

8. Fallbacks.

In cases when no glyph could be constructed, Cedilla will fall back on ever less satisfactory alternatives. For example, a Polish or German opening quotation mark will first be replaced by a sequence of two single quotation marks which will in turn be replaced with a sequence of two commas.

The fallbacks mechanism is a rather complex beast, and interacts in wonderful and mysterious ways with verious mechanisms for glyph generation (notably compositing). Currently, the only place where it is described is the source code.

INSTALLING CEDILLA

Please see the file 'INSTALL' for information on installing Cedilla.

INVOKING CEDILLA

Please see the 'cedilla(1)' manual page for information on invoking cedilla.

CONFIGURING CEDILLA

Cedilla is configured by inserting fontset definitions in a file named 'cedilla-config.lisp' which, by default, lives in '/etc/'. A fontset definition is an invocation of the macro DEFINE-FONTSET with two parameters: the name of the fontset and an ordered list of font definitions.

```
(define-fontset "times"
  `((:afm "ptmr.afm" :omit ("mu"))
    (:afm "psyr.afm")
    (:afm "pzdr.afm" :encoding ,#'zapf-dingbats-encoding)
    (:built-in :width 667 :height 662 :x-height 448)))

(define-fontset "utopia"
  `((:afm "UTRG____.afm" :resources ("UTRG____.pfa"))))
```

Every font definition is a list the first element of which is the font's type.

If the type is :AFM, then the entry specifies a font described by its AFM file. The font definition is of the form

```
(:AFM filename . options)
```

where filename is the location of the AFM file describing the font, and options is an alternating list of keyword/value (a 'plist' in Lisp parlance).

Options supported for the :AFM font type include:

- :RESOURCES : specifies a list of resources that must be downloaded if the font is to be used; currently, the only type of downloadable resource supported are fonts in either PFA or PFB format;

- :ENCODING : specifies that the font does not follow the Adobe glyph naming guidelines; the argument is a function that maps normal CCS to glyph names.

- :FIXED-ENCODING : specifies that the font is hopelessly broken, and should never, ever be reencoded; the argument is a function that maps normal CCS to font indices.

- :OMIT : specifies a list of names of glyphs that should be ignored; a typical application is to exclude the glyph “mu” that many Latin fonts include for compatibility with legacy encodings.

If the type is :BUILT-IN, then the entry specifies Cedilla's built-in font. The entry is of the form

```
(:BUILT-IN . options)
```

where options include:

- :WIDTH : specifies the width of typical glyphs in the font in units of one thousandth of the font size. The width of the capital Latin letter “C” in the principal font of the fontset is usually a good choice for this value.

- :FIGURE-WIDTH : specifies the width of digits and monetary symbols in the font. In not specified, defaults to the value of :WIDTH.

- :CAP-HEIGHT : specifies the capital height of the glyphs in the font. The height of the capital Latin letter “H” in the principal font is usually a good choice for this value.

• :X-HEIGHT : specifies the height of small letters in the font. The height of the small Latin letter “x” in the principal font is usually a good choice for this value.

If the type is :SPACE, then the entry specifies a fake font that provides a space glyph; this is meant for use with fonts that do not contain a useable space, such as some fonts designed for TeX; another application is to provide a space that has a different size from the one included in the font. The entry is of the form

(:SPACE . options)

where the only defined option is :WIDTH, the argument to which is an integer specifying the width of the space that the font will provide.

EXTENDING CEDILLA

Cedilla uses data from the UnicodeData file and from the Adobe Glyph List. These data are are supplemented with a table of (good) character alternatives, a table of (bad) character fallbacks, and a table of alternate glyph names. See the comments at the beginning of ‘unicode.lisp’ for information about the data structures used, and the file ‘data-add.lisp’ for the addtional data provided.

The built-in font is defined in the file ‘built-in.lisp’. You should feel free to add glyphs to this font, as Cedilla will only download those glyphs that are needed for the current print job.

ACKNOWLEDGEMENTS

Markus Kuhn coined the expression ‘best-effort typesetting’.

Local variables: ***
Mode: text ***
Coding: utf-8 ***
fill-column: 76 ***
End: ***