L2/19-385

Unihan: Publish single-property files

Markus Scherer, 2019-dec-02

Status quo

We publish CJK character properties as a collection of Unihan*.txt files, each with multiple properties. For example, Unihan_IRGSources.txt has data for kIICore, kIRG_GSource, and others. (See UAX #38.)

We actually publish a single Unihan.zip file of that .txt file collection.

Proposal

I propose that we split the Unihan data into single-property files, such as kIICore.txt, and publish a .zip file of those. Suggested filename: UnihanSingles.zip

We may or may not remove the Unihan.zip file; we may keep both for a release or two.

(Keeping both .zip files would be similar to how we publish both LineBreak.txt and extracted/DerivedLineBreak.txt, or how we have the General_Category values in UnicodeData.txt as well as separately in extracted/DerivedGeneralCategory.txt.)

The set of single-property files is subject to change. Many Unihan properties are provisional, so properties come and go, and thus files will come and go. A missing file is a more obvious indication of a removed property than the mere absence of data where it used to be.

File format

I propose that we name each file for its property, such as kTotalStrokes.txt.

Each file should have data lines with two fields, as is customary for UCD single-property data files: The code point or range of code points, and the value. We should use a TAB as field separator, not a semicolon, because some of the Unihan properties use or allow semicolons as part of their values.

Comment lines should be allowed as usual.

I have a small, publicly available script that reads the Unihan*.txt files and writes the single-property files:
<Unicode Tools>/py/splitunihan.py

Example: These are the first few lines of kTotalStrokes.txt:

```
3400	5
3401..3402	6
3403..3404	3
3405	2
```

```
3406    6
3407..3409      3
340A..340B     4
340C..340F     5
```

Rationale

The Unihan*.txt files are large and verbose. Reading just a few properties requires scanning past a lot of data for other properties. The single-property .txt files are each relatively small, and in total they are much smaller than the Unihan*.txt files (15MB vs. 35MB). In compressed form (.zip), the single-property files are significantly smaller as well (4.6MB vs. 6.4MB).

More importantly, it is neither obvious nor stable which Unihan*.txt file contains the data for which property. For example, kIICore and kUnihanCore2020 are in different files, Unihan_RadicalStrokeCounts.txt does not have data for kRSUnicode, and Unicode 13 moves kTotalStrokes from Unihan_DictionaryLikeData.txt to Unihan_IRGSources.txt. Moving a property from one file to another is very disruptive for parsers.

When I asked about the move of kTotalStrokes, I was advised that “the robust way to handle” Unihan data is to concatenate the Unihan*.txt files and to filter desired properties back out of that, before parsing the resulting subset of the data. This is not explicitly documented. Essentially, we expect users of Unihan data to discover the instabilities and come up with their own ways of dealing with them.

Let’s publish the data in a form that is more easily usable.

PS

I have switched the UCD maintainers’ “Unicode Tools” to using the proposed single-property Unihan .txt files.