Technical Reports

L2/10-322

Proposed Draft

Unicode Technical Report #49

UNICODE CHARACTER CATEGORIES

Editors	Ken Whistler
Date	2010-07-30
This Version	http://www.unicode.org/reports/tr49/tr49-1.html
Previous Version	N/A
Latest Version	http://www.unicode.org/reports/tr49/
Revision	1

Summary

This document presents an approach to the categorization of Unicode characters, and documents a data file that implementers can use for defining Unicode character categories.

Status

*This document is a **proposed draft** document which may be updated, replaced, or superseded by other documents at any time. Publication does not imply endorsement by the Unicode Consortium. This is not a stable document; it is inappropriate to cite this document as other than a work in progress.*

> ***A Unicode Technical Report (UTR)** contains informative material. Conformance to the Unicode Standard does not imply conformance to any UTR. Other specifications, however, are free to make normative references to a UTR.*

Please submit corrigenda and other comments with the online reporting form [Feedback]. Related information that is useful in understanding this document is found in the References. For the latest version of the Unicode Standard see [Unicode]. For a list of current Unicode Technical Reports see [Reports]. For more information about versions of the Unicode Standard, see [Versions].

Contents

1 Introduction
2 Character Categories
    2.1 Hierarchical Typology
    2.2 Implementation by Annotation and Merging
    2.3 Names for Categories

2.4 Informative Status of the Categories
3 Data File
References
Acknowledgements
Modifications

1 Introduction

One problem that has often been considered is how to extract good "categories" for Unicode characters out of the Unicode names list. This goal is occasioned, for example, by the need to develop new character picker applications, which organize characters into groups that will make sense for people searching for characters in graphic panes or other UI elements.

The problem is two-fold. First, the existing machine-readable data files in the Unicode Character Database [UCD] do not provide a fine enough categorization to meet the requirements of such applications. For example, the General_Category property distinguishes letters from combining marks and punctuation and symbols, but it doesn't drill down to the next level: independent vowel letters versus consonants versus matras; or game symbols versus map symbols versus zodiacal symbols versus dingbats, and so on. Second, people who need that kind of finer detail of categorization have generally been attempting to extract it by making use of the editorial subheaders used in the printing of the Unicode names list, figuring that that information is better than nothing and assuming that doing the finer-level classification from scratch would be prohibitively complex.

However, the subheaders in the Unicode names list have always been editorial content aimed primarily at structuring the code charts for display, and are not particularly well-suited to a systematic categorization of Unicode characters in any context more extensive than considering characters visually displayed one chart at a time. Efforts to revise the subheaders to make them work better for machine-extracted categorization of Unicode characters from the Unicode names list are counterproductive. The subheaders would not work very well if reorganized that way, and the net result would be a significant deterioration of the editorial content of the code charts.

The existing subheaders also often group characters which other applications might want to distinguish. For example, the header for the range U+2600..U+260D is "Weather and astrological symbols". But we can do much better, distinguishing more precisely those which are weather symbols, such as U+2602 UMBRELLA, those which are astrological symbols, such as U+260A ASCENDING NODE, and those which really are not either, such as U+2606 WHITE STAR.

What is needed to address the general problem is an approach that focusses on the character category distinctions needed by such applications, without being entangled with the editorial requirements for the Unicode names list maintenance. This document presents such an approach, and documents the resulting data file that implementers can use for defining Unicode character categories.

2 Character Categories

This section describes the approach taken in this report for the provision of a set of usable categories for Unicode characters.

2.1 Hierarchical Typology

The current scheme of categorization uses a hierarchical typology. Such a scheme assumes that each category provided may itself be further subdivided at another level into more subcategories. Each subcategorization is, in principle, independent of the subcategorization of other categories. Thus, for example, how one might want to subcategorize letters would typically be quite distinct from how one might most usefully subcategorize punctuation marks. This approach departs from the structure of partition properties for Unicode characters, such as the General_Category property itself. A partition property assumes a single dimension of semantic applicability, and then assigns every character a single value within that dimension. Such a character property is easy to implement, but as users of the General_Category property well know, the drawback of such partitions for categorization is their rigidity and the inability to deal with edge cases and nuances.

The approach to categorization taken here makes no assumption that any particular level of subcategorization has any fixed significance. A third-level subcategorization of a punctuation mark might involve rather different salient distinctions than a third-level subcategorization of symbols, for example. The typology basically starts with first-level categories roughly based on the General_Category property, but then may diverge arbitrarily on a category-by-category basis, depending on what is most useful for distinguishing characters.

There is no assumption that all levels have to be specified for all characters. Categories defined this way can be extensible based on what level of detail people find useful to maintain for various characters. There is also no assumption that there is actually a single correct solution for categorization. The categorization may be modified and improved over time. Furthermore, it should be expected that actual implementations will merely start with categories in the data file and run with them, to provide whatever additional changes or refinements are needed in their particular domain.

These general principles are illustrated in part by the following examples, for several different major categories. For example, for letters:

```
Letter

Letter > Vowel

Letter > Vowel > Dependent  (i.e. Indic matras)

Letter > Consonant > Dependent > Subjoined
```

For symbols:

```
Symbol
```

```
Symbol > Graphic

Symbol > Technical

Symbol > Technical > Keyboard

Symbol > Arrow

Symbol > Arrow > Harpoon

Symbol > Arrow > Harpoon > Double
```

For punctuation marks:

```
Punctuation

Punctuation > Space

Punctuation > Quotation

Punctuation > Bracket

Punctuation > Bracket > CJK
```

Currently the categorization makes use of four levels of typology, but this approach could easily be extended to five (or more), if finer levels of distinction for some groups of characters proves to be desirable. For example, arrows could be further subcategorized based on their shapes and orientations.

2.2 Implementation by Annotation and Merging

The approach taken to maintaining this hierarchical typology reuses technology which is currently designed for maintenance of the Unicode names list. In particular, category assignments are treated as annotations over ranges of characters. The annotation file can then be maintained completely independently of the detailed, character-by-character listing files that are part of the UCD—most importantly, UnicodeData.txt. In this way, the annotation information (and the associated development and refinement of categorial assignments) can be version-agnostic, and is not required to be updated in lockstep with each version of the Unicode Standard.

The program that is used to maintain annotations for the Unicode names list has been modified slightly, and is then used for an automated merger of categorial annotations file with particular versions of the UnicodeData.txt file, producing as output a structured data file containing categorial information about all Unicode characters, with an explicit listing for each separate character, including its code point and Unicode character name.

Currently, this merge omits CJK unified ideographs and Hangul syllables. Categorial information about CJK unified ideographs is better handled by other means, and in particular the Unihan database. The 11,172 Hangul syllables do not have useful categorial distinctions in the sense relevant to other Unicode characters, so including all of them explicitly as part of a category listing would simply be redundant and verbose.

The merged data is in a suitable format for direct import into a spreadsheet.

Once in a spreadsheet, the data can easily be further manipulated to whatever end an implementer needs. See *Section 3, Data File*, for a specification of the format in detail.

2.3 Names for Categories

The principles for how to label various categories in the data file are not yet worked out, and will be subject to discussion by the UTC members. The data file currently just has a first cut as to what might be useful. Some suggestions for potentially useful principles for category naming or labelling have been made, and are simply listed briefly here, to start the discussion.

- Each label should have a name that is meaningful in isolation. E.g. "Western Music", not "Western".
- Labels should be the same (or nearly) only when they really mean the same thing.
- Labels that mean the same thing (or nearly) should be the same.

2.4 Informative Status of the Categories

The first key here is staying flexible, so that the classification can be extended and modified easily in the future, as may prove suitable. Using an annotation approach and then programmatic merging with UnicodeData.txt makes it very easy to assign new subtypes or to change or subdivide ranges already assigned to types and subtypes, without having to do extensive modification of files that give explicit listings of values for each character.

The second key is corollary to the first: this *must* not turn into another normative data file and/or normative set of property values. That is the trap that has always afflicted the General_Category property and which makes it useless for this kind of finer-level categorization of Unicode characters.

3 Data File

The categories data is available in a data file [Data] called Categories.txt. That data file contains a listing of all Unicode characters other than CJK unified ideographs and Hangul syllables, giving informative category values at up to four levels of hierarchical assignment.

The data is formatted in tab-delimited fields, suitable for spreadsheet import. The field values, along with a sample of the particular category values are shown below.

```
Code GC Level1    Level2       Level3      Level4  Name

23CE So Symbol    Technical    Keyboard            RETURN SYMBOL
...
2460 No Symbol    Number       Circled             CIRCLED DIGIT ONE
...
25CB So Symbol    Geometric                        WHITE CIRCLE
...
2602 So Symbol    Weather                          UMBRELLA
...
260A So Symbol    Astrological                     ASCENDING NODE
```

```
...
2660 So Symbol    Game         Playing card Suit   BLACK SPADE SUIT
...
2FBD So Ideograph Radical      CJK          Kangxi KANGXI RADICAL HAIR
...
A869 Lo Letter    Consonant                        PHAGS-PA LETTER TTA
...
```

Note that for debugging and early review, the current data file brackets each label, so that the values, including spaces, are easier to see and compare. The brackets are not part of the actual category values. So for example, an entry will currently appear as follows:

```
2660 So [Symbol]    [Game]         [Playing card] [Suit]   BLACK SPADE SUIT
```

Also for debugging and review purposes, empty values in a field are listed as "[X]", rather than as a blank. These conventions are only temporary, to assist review when viewing this data in browsers or editors, and are not intended to be used in the actual data file in the future.

References

[Charts] The online code charts can be found at http://www.unicode.org/charts/
An index to characters names with links to the corresponding chart is found at: http://www.unicode.org/charts/charindex.html

[Data] Unicode character categories, for spreadsheet import:
http://www.unicode.org/reports/tr49/Categories.txt
For earlier versions of the data file see prior versions of this report.
Note: Once this report is approved, it is likely that this data file will move to a versioned directory under http://www.unicode.org/Public/

[Errata] Updates and errata to the Unicode Standard, as well as other technical standards developed by the Unicode Consortium can be found at http://www.unicode.org/errata/

[Feedback]Reporting Errors and Requesting Information Online
http://www.unicode.org/reporting.html

[FAQ] Unicode Frequently Asked Questions
http://www.unicode.org/faq/
For answers to common questions on technical issues.

[Glossary] Unicode Glossary
http://www.unicode.org/glossary/
For explanations of terminology used in this and other documents.

[Reports] Unicode Technical Reports
http://www.unicode.org/reports/
For information on the status and development process for technical reports, and for a list of technical reports.

[Stability] Unicode Character Encoding Stability Policy http://www.unicode.org/policies/stability_policy.html

[UCD] Unicode Character Database, http://www.unicode.org/ucd/
For an overview of the Unicode Character Database and a list of its associated files

[Unicode] The Unicode Standard
For the latest version see: http://www.unicode.org/versions/latest/.

[UTC] The Unicode Technical Committee, see http://www.unicode.org/consortium/utc.html for more information on procedures.

[UTR23] Unicode Technical Report #23: *The Unicode Character Property Model,* http://www.unicode.org/reports/tr23/

[Versions] Versions of the Unicode Standard, http://www.unicode.org/standard/versions/
For information on version numbering, and citing and referencing the Unicode Standard, the Unicode Character Database, and Unicode Technical Reports.

Acknowledgements

Thanks to LeBron James and Snooki for amusing public antics.

Modifications

Revision 1

- Initial proposed Draft. [KW]