L2/014-022
Date: 2014-01-12

Title: Character Name considerations
Source: Michel Suignard
Distribution: UTC

Summary: The following document describes character names as used in both ISO/IEC 10646 and Unicode. It also contains a case study of the names proposed for the Tangut repertoire and suggests an alternate solution for these. This work would not have been possible without the detailed analysis done by the Tangut experts in their proposal (WG2 N4522 and 4525).

1. Character name status as defined by ISO/IEC 10646 and Unicode:

1.1 General

Character name is an essential part of the definition of an encoded character for both standards. In many cases the name provides an easy way to identify the character by being descriptive and semantically defined. For example, in 0041 LATIN CAPITAL LETTER A the name 'LATIN CAPITAL LETTER' provides the information that the character belongs to the Latin script, is a capital letter, and is commonly associated with the letter 'A'. Although ISO/IEC 10646 and Unicode share the same character names for all encoded code points they differ significantly in their definition of the character names.

1.2 ISO/IEC 10646 name status

For this standard the name is 'assigned' to each graphic and format [encoded] character. Although the standard refers normatively to character properties in the Normative References clause when using the General Category (Gc), Bidi_Mirrored and others, the character name is never defined as such. It is however explicitly made immutable. Follows an extract of the pertinent text in the latest version of ISO/IEC 10646 (4th edition under DIS ballot).

> **6.4 Naming of characters**
>
> This International Standard assigns a unique name to each graphic and format character. The name of a character either
>
> a) denotes the customary meaning of the character, or
> b) describes the shape of the corresponding graphic symbol, or
> c) follows the rule given in 24.6 for Chinese /Japanese/Korean (CJK) ideographs, or
> d) follows the rule given in 24.7 for Hangul syllables.
>
> Some characters may have one or more alternate names called character name aliases which are correction of the original names. Additional rules to be used for constructing the names of characters are given in 24. …
>
> **7 Revision and updating of the UCS**
>
> …
> The names and code points allocation of all characters in this coded character set shall remain unchanged in all future editions and amendments of this standard. …

1.3 Unicode Status

For Unicode, the character name is a normative immutable property. Consequently, the standard includes a formal definition for that property. It also provides some User Interface considerations. The following is an extract of the Chapter 4 Character Properties from the Unicode Standard 6.2.

4.8 Name

Unicode characters have names that serve as unique identifiers for each character. The character names in the Unicode Standard are identical to those of the English-language edition of ISO/IEC 10646.

Where possible, character names are derived from existing conventional names of a character or symbol in English, but in many cases the character names nevertheless differ from traditional names widely used by relevant user communities. The character names of symbols and punctuation characters often describe their shape, rather than their function, because these characters are used in many different contexts. …

Stability. Once assigned, a character name is immutable. It will never be changed in subsequent versions of the Unicode Standard. Implementers and users can rely on the fact that a character name uniquely represents a given character. …

Character Name Aliases. Sometimes errors in a character name are discovered after publication. Because character names are immutable, such errors are not corrected by changing the names. However, in some limited instances (as for obvious typos in a character name), the Unicode Standard publishes an additional, corrected name as a normative character name alias. (See Definition D5 in Section 3.3, Semantics.) Character name aliases are immutable once published and are also guaranteed to be unique in the namespace for character names. A character may, in principle, have more than one normative character name alias. …

Unicode Name Property

Formally, the character name for a Unicode character is the value of the normative character property, “Name”. Most Unicode character properties are defined by enumeration in one of the data files of the Unicode Character Database, but the Name property is instead defined in part by enumeration and in part by rule. A significant proportion of Unicode characters belong to large sets, such as Han ideographs and Hangul syllables, for which the character names are best defined by generative rule, rather than one-by-one naming.

Formal Definition of the Name Property. The Name property (short alias: “na”) is a string property, defined as follows:

- For Hangul syllables, the Name property value is derived by rule, as specified in Section 3.12, Conjoining Jamo Behavior, under “Hangul Syllable Name Generation,” by combining a fixed prefix and appropriate values of the Jamo_Short_Name property. For example, the name of U+D4DB is hangul syllable pwilh, constructed by concatenation of “hangul syllable” and three Jamo_Short_Name property values, “p” + “wi” + “lh”.

- For ideographs, the Name property value is derived by concatenating the string “cjk unified ideograph-” or “cjk compatibility ideograph-” to the code point, expressed in hexadecimal, with the usual 4- to 6-digit convention. For example, the name of U+4E00 is cjk unified ideograph-4e00. Field 1 of the UnicodeData.txt data file uses a special convention to indicate the ranges of ideographs for which the Name property is derived by rule.
- For all other Graphic characters and for all Format characters, the Name property value is as listed in Field 1 of UnicodeData.txt. For example, U+0A15 gurmukhi letter ka or U+200D zero width joiner.
- For all other Unicode code points of all other types (Control, Private-Use, Surrogate, Noncharacter, and Reserved), the value of the Name property is the null string. In other words, na=“”.

The generic term “character name” refers to the Name property value for an encoded Unicode character.
…

User Interfaces. A list of Unicode character names may not always be the most appropriate set of choices to present to a user in a user interface. Many common characters do not have a single name for all English-speaking user communities and, of course, their native name in another language is likely to be different altogether. The names of many characters in the Unicode Standard are based on specific Latin transcription of the sounds they represent. There are often competing transcription schemes. For all these reasons, it can be more effective for a user interface to use names that were translated or otherwise adjusted to meet the expectations of the targeted user community. By also listing the formal character name, a user interface could ensure that users can unambiguously refer to the character by the name documented in the Unicode Standard.

2. A study of character names currently specified

In a first approach, character names can be seen as either based on rules (CJK Ideographs and Hangul Syllables) or based on semantic/graphical. However, it is more subtle:

- Some rules (for example Hangul Syllables) are creating names that are semantic in nature. But the encoding made possible to derive those names from the code points. The rule is just a shortcut to express the same semantic content.
- Most of the rules incorporate in the name a constant part (for example CJK UNIFIED IDEOGRAPH) with the code point of the character; example: 4E00 CJK UNIFIED IDEOGRAPH-4E00. Their true identification is done by source references.
- Some names, although not rules based, have no semantic. For example, Egyptian Hieroglyphs (not rules based) have names based on catalog numbers which are analogous to source references. Many other repertoires use the same catalog numbers schemes in their name: Linear A, Anatolian Hieroglyph, and Ancient Greek Musical Notation.
- Some names are a mix of catalog number and semantic (many examples in the Linear B Syllabary block), example: 10043 LINEAR B SYLLABLE B071 DWE.
- Some rule based characters (CJK Compatibility ideographs) still have their names contained in the repository for non-rule based characters. This is because originally these character names were not defined as rule based.

- Symbols can either be described semantically or graphically. There is no definitive convention. Compare for example 2610 ☐ BALLOT BOX with 25A1 □ WHITE SQUARE.

The immutability of character names is not problematic for names that are describing meaning/semantic or graphic shapes. Even a typo does not mask the identity of the character, and in the most nefarious cases, character name aliases can be added. However, it becomes problematic for source information such as catalog numbers. The issue was avoided altogether for CJK Ideographs by not including the source reference in the name. Source references, although normative, are not immutable and can be modified. However if the names were to include such erroneous references, they cannot be fixed and any tools mining that information have to build exception tables. In other words, if a catalog number is found to be in error in a name, it cannot be fixed and the character name and the formal catalog number reference will be erroneous forever.

Another less than optimal case is the semantic/catalog hybrid name. Typical examples are the Linear B Syllabary and Linear B Ideograms blocks with character names as 10043 LINEAR B SYLLABLE B071 DWE, 100C3 LINEAR B IDEOGRAM B191 HELMET, containing catalog numbers (071 and 191), and descriptions (DWE and HELMET). Such hybrid names creates an unnecessary complication to extract either the catalog or the semantic portions.

Related to this hybrid case, it is difficult to merge various catalog schemes in a single name. By having a syntax restricted to a small repertoire (A-Z, 0-9, Space, Hyphen-Minus), names are ill suited to describe multiple references. Typically you want a syntax allowing multiple fields per line (such as comma separated format or similar).

While these issues are somewhat workable on small blocks they become very problematic in large repertoires. In all cases, using catalog/source references in names is not a good idea and should be avoided for all future repertoires, whatever the size of the repertoire.

3. Case study: Tangut repertoire

Tangut is a proposed repertoire with currently 6125 characters which is using names with catalog numbers. These numbers are extracted from several catalogs with their own convention. These are the following (from WG2 N4522):

- **Lǐ Fànwén 2008** (*Tangut-Chinese Dictionary*, 2nd ed.)
- **Kyčanov and Arakawa 2006** (*Tangut-Russian-English-Chinese Dictionary*)
- **Lǐ Fànwén 2006** (Comparative Study of *Wuyin Qieyun* and *Wenhai Baoyun*)
- **Hán Xiǎománg 2004** (PhD dissertation on the correct form of Tangut ideographs)
- **Lǐ Fànwén 1997** (*Tangut-Chinese Dictionary*, 1st ed.)
- **Lǐ Fànwén 1986** (*Study of the Homophones*)
- **Sofronov 1968** (*Grammar of the Tangut Language*)
- **Nishida 1966** (*Little Dictionary of Tangut*)
- Shǐ Jīnbō *et al*. 1983 (*Study of the Sea of Characters*)

Initially in 2008, Tangut had been proposed with names similar to the CJK schema such as:
xxxxx TANGUT CHARACTER-xxxxx (xxxxx being the proposed code point for the characters) . This was presented in WG2 N3509 (Working Draft ISO/IEC 10646 2nd edition 2008), in sub-clause 6.4:

> ISO/IEC 10646 assigns a unique name to each character. The name of a character either
>
> a. denotes the customary meaning of the character, or

b. describes the shape of the corresponding graphic symbol, or
c. follows the rule given in 24.5 for Chinese /Japanese/Korean (CJK) ideographs, or
d. follows the rule given in 24.6 for Tangut characters, or
e. follows the rule given in 24.7 for Hangul syllables.

And in sub-clause 24.6:

> **24.6 Character names for Tangut characters**
> For Tangut characters the names are algorithmically constructed by appending their coded representation in hexadecimal notation using their five hexadecimal digit value to "TANGUT CHARACTER-".

The proposal never progressed enough to address any issue concerning referencing of the sources as would have been expected. But all technical changes required for the repertoire had been incorporated in the working draft.

More recently (October 2012, revised in January 2014), a newer proposal was made proposing roughly the same repertoire with new names incorporating seven sources (two of the source above are not used in the names). Syntax is as following (WG2 N4522 and 4525), with the sequence in bold following 'TANGUT SIGN':

- Lǐ Fànwén 2008 : **L2008-*nnnn*** (where a character maps to a single entry) or **L2008-*nnnn*-*nnnn*** (where a character maps to two entries)
- Lǐ Fànwén 2006 : **L2006-*nnnn***
- Hán Xiǎománg 2004 : **H2004-*nnnn***
- Lǐ Fànwén 1997 : **L1997-*nnnn***
- Lǐ Fànwén 1986 : **L1986-*nnnn***
- Sofronov 1968 : **S1968-*nnnn***
- Nishida 1966 : **N1966-*nnn*-*nnn*** (the last digit may be a letter or a double letter)

For example, the name list entry for 1701C reads: TANGUT SIGN L2008-0042-4537.

This schema presents some issues:

- It conflates in a single property a rather inefficient way the source name and multiples source values,
- It also mixes in a similar notation either a dual source or a single source with two elements,
- It merges in a single name space (the character name) all the 7 sources,
- And worse of all, because the names are immutable, errors cannot be fixed on what is a data intensive notation. Furthermore, the single name space for the 7 sources can never be dis-unified, because of immutability (i.e. it is not a possible to ever create a notation allowing more than one source in the name).

Guaranteeing an error free list of 6125 items is a tall order. The CJK source references share a similar issue and errors are frequently spotted, more than ten years after these references were originally registered. Because these references are not immutable, it is manageable.

In fact, a close inspection of the current Tangut name list seems to indicate some issues:

a) Case of 184A1

The document WG2 N4525, page 64, shows the following value for 184A1:

184A1 𘒡 TANGUT SIGN L2006-5597

But in WG2 N4522, page 605, it shows (other sources cut to fit in page):

Code Point	Glyph	Nominal IDS	Radical	Stroke Count	Stroke Order	Lǐ Fànwén 2008 (Xià-Hàn Zìdiǎn)	Lǐ Fànwén 2006	Lǐ Fànwén 1997 (Xià-Hàn Zìdiǎn)	Lǐ Fànwén 1986 (Tóng Yīn)
184A0	𘒠	[illegible]	R285	19	GABBBACCCQ DCABABFAA	𘒠 1186	𘒠 5997		
184A1	𘒡	[illegible]	R285	20	GABBBACCCQ DCABABFAAC		𘒡 1186	𘒡 1186	𘒡 5085

According to the source used (L2006), the value should have been 1186, not 5597. Note that 184A0 has L2008-1186, but it is a different source (L2008 instead of L2006).

Then, there is the character 17C06:

17C06 𗰆 TANGUT SIGN L2008-5597

with 5597 as its source value for L2008. However, according to WG2 N4522:

Code Point	Glyph	Nominal IDS	Radical	Stroke Count	Stroke Order	Lǐ Fànwén 2008 (Xià-Hàn Zìdiǎn)	Lǐ Fànwén 2006	Lǐ Fànwén 1997 (Xià-Hàn Zìdiǎn)
17C06	𗰆	[illegible]	R098	15	DCJCDABCCC QEAMC	𗰆 5597	𗰆 5597	𗰆 5597

It also has the same source value for L2006 and L1997, meaning it could have been encoded as 'TANGUT SIGN L2006-5597' if L2008-5597 did not already exist. Clearly an error here for 174A1 with a probable fix for the name to be TANGUT SIGN L2006-1186 (instead of TANGUT SIGN L2006-5597).

b) **The case of H2004 sources**: All sources referenced using H2004-nnnn from 'Hán Xiǎománg 2004' use a H2004 notation not distinguishing between the H1 and H2 sub-division which corresponds to the two lines for that column as seen below:

Code Point	Glyph	Nominal IDS	Radical	Stroke Count	Stroke Order	Lǐ Fànwén 2008 (Xià-Hàn Zìdiǎn)	Lǐ Fànwén 2006	Lǐ Fànwén 1997 (Xià-Hàn Zìdiǎn)	Lǐ Fànwén 1986 (Tóng Yīn)	Kyčanov & Arakawa 2006	Hán Xiǎománg 2004
1744E			R040	11	DCACCCQABEA	4989	4989	4989	0977	1038	1012 1063
1744F			R040	11	DCACCCQCCCQ	4950	4950	4950	0978	2928	1013 1064
17450			R040	11	DCACCCQDCCB	5002	5002	5002	0975	0444	
17451			R040	11	DCACCCQDCCQ						1015 1066

H1 and H2 cannot share the same numeric values because their name spaces collide, for example, the character 17CEB has H1=3000 and H2=3217 and 17DC4 has H1=3217 and H2=3337 (source WG2 N4522). It is just because they both have a primary L2008 source and therefore do not use the H2004 source value that the issue is avoided.

Furthermore, as of now, all 9 proposed characters using that source are from the H1 subset. It seems however an oversight to not carry that distinction in the source reference. Should an extension to the repertoire reveals a character only present in the H2 subset, the current schema would not carry over.

These two issues were found doing a quick check of the repertoire, so there is a great chance that more errors exist. This does not pretend to be a thorough review of the repertoire but just a quick demonstration on how using cataloging on a large repertoire is prone to error and therefore including catalog number in an immutable property (name) is not a wise idea.

4. Case for catalog name based made by document WG2 N4522

The document WG2 N 4522 in page 43 provides some elements of discussion concerning Tangut name. For example, section 9.0 ‘Character Names’, first paragraph:

> According to ISO/IEC 10646 clause 6.4 "Naming of characters", the name of a character (other than CJK ideographs and Hangul syllables) should denote its customary meaning or describe the shape of the corresponding graphic symbol.

While it describes the a) and b) sub-clause of 6.4, the text omits to present the c), d) sub-clause describing the rule based for other characters (CJK ideographs and Hangul syllables), described here in section 1.2. Obviously, should the Tangut been encoded, a new e) sub-clause could be added, along with any appropriate text required for a new repertoire.

Then the second paragraph says:

> In a previous Tangut encoding proposal by Richard Cook (N3297), it was proposed to give Tangut characters algorithmic names of the pattern "TANGUT CHARACTER-17000", although such names contravene ISO/IEC 10646 clause 6.4, and no rationale for this naming convention was provided.

It is a rather incorrect characterization of the situation. Saying that new repertoire names (and any other technical change induced by an amendment) contravene existing text of a standard is absurd. It is the whole point of an amendment to make the appropriate technical changes to the standard. In the first proposed amendment (and the corresponding working draft of the standard) introducing the Tangut repertoire, a new sub-clause had been introduced to add the new rule based names along with CJK ideographs, and Hangul syllables. This is a normal process and does not constitute a violation or contravention.

Then in section 9.1 'Rationale for Using Algorithmic Names', the points made for using rule-based name are size and error prone. As shown above, the error concern is real and already demonstrated today as shown above.

The other points that are not mentioned in the document in favor of a rule based approach are the following:

- By dis-associating the cataloging from the name, it allows separate, data-mining friendly access to these various catalog numbers,
- It does not force catalog numbers to be frozen by an immutable scheme (name),
- It is expandable as new needs arise or more research is performed to show additional information,
- It allows a richer presentation in the name, including radical stroke info and multiple sources if so desired,
- It can do anything that the rationale for using catalog-based Names (section 9.2) shows with a very simple processing of an additional file. And that additional file may contain much more information that any name based system can ever do.

Then in the second paragraph of section 9.3 'Discussion' the following point is made:

> We are also concerned about the implications of using algorithmic names for Tangut characters for future additions of new sets of characters to the standard. We consider that using algorithmic names for Tangut characters is an architectural change to the standard is[sic] it necessitates a change to ISO/IEC 10646 clause 6.4, and may impact on future character encoding proposals. If algorithmic names for Tangut were to be accepted, what would the criteria be for using algorithmic names versus meaningful names be for future character additions? Is there some, as yet unspecified, threshold for the size of the proposed character set (e.g. all future proposed character sets with more than 1,000 characters should use algorithmic names)? Or would the nature of the proposed character set have a bearing on whether to use meaningful or algorithmic names (e.g. all large, historic scripts should use algorithmic names)? Or would the nature of the proposed character names be relevant to the issue (e.g. algorithmic names are preferable to catalogue-based names)? We believe that these are questions that need answering, and that the issue of whether to use algorithmic names for Tangut characters should be framed in the larger context of character naming rules in the ISO/IEC 10646 standard.

The point made that adding a sub-clause to an existing set of assertions (clause 6.4) could be an architecture change has been already answered. It is no more an architecture change than adding any repertoire with the necessary text adjustment in the standard. The other points deserve study and this document is a partial answer to that. It does provide an alternate solution for Tangut which can be used for Tangut and similar repertoire in the pipeline. It could also be used for existing repertoires that have non rules based names. This is something that was done for CJK Compatibility Ideographs and it could be done for other existing repertoires.

Finally the document WG2 N4522 makes a point about applications as follows (section 9.2 ‘Rationale for Uisng Catalog-based names’, page 43, 3rd item:

> Third-party character map and character picker applications normally allow users to find a given Unicode character by character name, and having algorithmic names would mean that users would be unable to search for individual Tangut characters in such applications unless they already knew the character's code point (it is unlikely that general-purpose character picker applications would support searching for Tangut names provided in a supplementary file in Unicode or ISO/IEC 10646).

It is trivial to build a pseudo-character name including such catalog information from a supplementary file. It could either be done directly in the character picker application or with an outside script in few lines of code. In addition, most character pickers already access additional files beyond the names list to determine blocks, scripts, and other criteria for selection. In fact, having a richer supplementary data file would allow the character picker to offer much more capability in accessing the characters, such as radical, stroke, multiple sources, etc… At the end, the user community will be much better served by having a richer data set made publicly available and easily exploitable by applications.

5. A better solution for the Tangut repertoire

Instead of trying to fit all sources in the names as currently proposed, it seems wiser to use a mechanism where the name is a mere index (for example using the code point) and put the source reference in a separate data file where it can still be a normative property but not immutable. This also allows to expand the data file to represent the sources in multiple columns if desired. This has the added benefit of allowing the representation of ancillary information such as the radical, stroke info, available in WG2 N4522 but not conveyed in the current name list. Analogous to what is done for CJK repertoire, these bits of information could be presented in the chart along with the character glyph.

Here is an example of how it could look in the code chart for the character 17E94..17E96:

17E94

 151.12

L2008-2352

17E95

 152.7

L2008-3255

17E96

 152.8

L2008-3256

The associated data file would contain the following info for these characters (syntax is tentative):

U+17E94	kRSTUnicode	151.12
U+17E94	kT_L2008	2352

U+17E95 kRSTUnicode 152.7
U+17E95 kT_L2008 3255
U+17E96 kRSTUnicode 152.8
U+17E96 kT_L2008 3256

For characters that have multiple source values for a single source such as 1701C, the data file may have the following content:
U+1701C kRSTUnicode 1.11
U+1701C kT_L2008 0042 4537

Then it becomes a presentation discussion whether the two sources should be shown in the chart or not. Some information can be made normative (sources/catalog), while others could be informative (radical, stroke count).

Some will recognize a similarity with the Unihan database format, but it just reflects a convenience factor for the tools. It does not imply any deep similarity between the CJK model and the Tangut model, just the recognition that both use sources or catalogs number and radical/stroke indexes.

The prototype assumes a single column model, where all source glyphs are merged but it can be easily extended with additional columns, should it be desired to show one glyph per source. This has no implication on immutable parts such as the character names. It is also possible to create a format where multiple source/catalogs are shown without additional glyphs.

Assuming 5 or 6 columns and 20 characters per column, 100 to 120 characters would be represented per chart pages, resulting in a more compact form of the repertoire than the current form which repeats unnecessarily the term ‘TANGUT SIGN’.

6. Conclusion

Names are not a good mechanism to convey source/catalog information which can be presented and stored using more flexible alternatives. Even for existing names which are currently created that way and therefore are immutable, it may worth considering creating these additional data storage model and modify the chart presentation accordingly.