- DRAFT - For UTC/L2 Consideration Only -

L2/98-042

Towards Formal Criteria on Disunification

Asmus Freytag, February 20, 1998

Background

There have been repeated proposals to disunify existing characters. These proposals cannot be fully evaluated without a more rigorous framework concerning the dis-unification/unification of characters. Without such formal criteria, all decisions are 'ad-hoc' and different proposals may get different level of review. UTC needs to spend some time in evaluating and possibly formalizing the criteria that we use do decide these cases. This is similar to the formalization we have done for script prioritization, but uses different criteria.

NOTE: The unification criteria used for the Han script are very thorough and quite sufficient. This document attempts to establish formal criteria for use in other scripts. There is no attempt to change the procedures used in Han unification.

What is dis-unification?

Disunification is the introduction of a new character which can also be encoded by an existing character. A strong case of disunification occurs where there is prevalent practice of using the existing character. A weak case of discunification occurs where there is little or no use of the existing character for the purpose for which the new character is intended.

> *Example* : Adding a Rongo-rongo period is a weak disunification if we assume that nobody has an existing implementation of Rongo-rongo. Adding a clone of a Latin letter for use with Cyrillic is a strong disunification as mixed Latin/Cyrillic character sets exist and have almost certainly been used for encoding the languages that the new characters are intended for.

Costs and Benefits

Proposals always claim that dis-unification brings a benefits. Formal criteria attempt to critically evaluate those benefits, but also compare them to the costs.Any disunification, but especially strong disunifications, introduce several types of cost to *all* complete implementations of the Unicode Standard.

- First, any complete implementation will have to add and support both an additional entry in the properties as well as an additional glyph, or glyph mapping for the disunified character.
- Second, whenever the character in question has no appearance distinction, there is the cost of accidental confusion and mis-identification. All implementations will need sophisticated handling of equivalences, especially, where disunification occurs on well-established characters (as opposed to among the characters of an entirely new script being fine- tuned in the proposal stage).
- Third, keyboards that support the disunification need to be widely (and by default) available, this is especially troublesome for strong disunifications of Latin characters as most keyboards have a Latin layer from which it is easy to type the existing and now disunified character.

Criteria of analysis

The following questions are designed to evaluate the costs associated with the disunification.

1. Is there a glyphic distinction?
2. Is the use of the new character restricted to a new context (e.g. use with a novel script).?
3. Is the use of the existing, ambiguous character instead of the proposed new character common, prevalent or established practice?
4. Does the character exist in ASCII?

Against this the benefits need to be counted. Not all will apply in each case. The following are questions designed to evaluate the claimed benefits.

- First, appearance: does disunification help to allow multilingual monofont text in an environment where this is commonly needed? In what way?
- Second, layout: does disunification solve common layout differences (this would mostly be true for punctuation)?
- Third, searching/sorting: Is there a *common* case where disunification allows better support for these?
- Fourth, mapping to another standard: Is there a widely used standard that disunifies the characters in question? Are the characters in question the *only* ones that prevent cross mapping?