In the slides of “coordination between C, J and K” presented in CDNC Shanghai meeting, ICANN gave some coordination principles about repertoire, variant mapping and variant types as follows:

* Each CJK panel creates an LGR
* Each LGR includes a repertoire and variants
* If an LGR includes Han characters:
* the variant MAPPINGS must agree for all three panels
* the variant TYPES may be different
* the repertoires may be different

The meaning of “Variant mappings to agree”

* If any LGR defines a variant pair A <--> B
  + ALL OTHER LGRs that contain A
    - must also contain B
    - must also contain the mapping A ---> B
  + ALL OTHER LGRs that contain B
    - must also contain A
    - must also contain the mapping B ---> A

If the LGR from my Panel does NOT use variants

* The variant mappings must still be defined
* The variant mappings must agree with all other LGRs
* The variant type can be set to BLOCKED

If the LGR from my Panel uses variants?

* The variant mappings must agree with all other LGRs
* The variant type can be set to either BLOCKED or ALLOCATABLE
* The variant types do not have to agree across LGRs.

After the discussion with IP, Japan representatives and Korean representatives during ICANN 51, we’d like to raise a question to the highlighted text above.

There are TWO different understanding about “Variant mappings to agree” and “The variant type can be set to BLOCKED”.

1. “Variant mappings” is referred to a loose coupled relationship, and “Variant Type” could be none ”allocatable” and none “blocked”, illustrated by the following example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Code Point** | **Allocatable Variant** | **Blocked Variant** | **Tag** |
| 一 (U+4E00) | -- | 壱 (U+58F1)  壹 (U+58F9)  弌 (U+5F0C) | und-hani |
| 壹 (U+58F9) | -- | 一 (U+4E00)  壱 (U+58F1)  弌 (U+5F0C) | und-hani |
| 弌 (U+5F0C) | 一(U+4E00) | 壹 (U+58F9)  壱 (U+58F1) | und-hani |
| 壱 (U+58F1) | 壹(U+58F9) | 一 (U+4E00)  弌 (U+5F0C) | und-hani |
| 一 (U+4E00) | -- | -- | und-Jpan |
| 壹 (U+58F9) | -- | -- | und-jpan |
| 弌 (U+5F0C) | -- | -- | und-jpan |
| 壱 (U+58F1) | -- | -- | und-jpan |

In this table, in Chinese language environment, “variants mappings” means those 4 code-points are exchangeable, for any given single code point of this “variant mappings“ family, the other code points could be either “Allocatable” or “Blocked”. But in Japanese environment, the exactly same 4 code points are all independent individual existence, they are not “allocatable” or “Blocked” for the other applicant, the allocation of one code point doesn’t affect the others.

1. Opposite to the case 1, the other understanding is that “Variant mappings” is a tight coupled relationship which applies in all LGRs, and “Variant Type” must be ”allocatable” or “blocked”, illustrated by the following example:

|  |  |  |  |
| --- | --- | --- | --- |
| **Code Point** | **Allocatable Variant** | **Blocked Variant** | **Tag** |
| 一 (U+4E00) | -- | 壱 (U+58F1)  壹 (U+58F9)  弌 (U+5F0C) | und-hani |
| 壹 (U+58F9) | -- | 一 (U+4E00)  壱 (U+58F1)  弌 (U+5F0C) | und-hani |
| 弌 (U+5F0C) | 一(U+4E00) | 壹 (U+58F9)  壱 (U+58F1) | und-hani |
| 壱 (U+58F1) | 壹(U+58F9) | 一 (U+4E00)  弌 (U+5F0C) | und-hani |
| 一 (U+4E00) | -- | 壹 (U+58F9)  弌 (U+5F0C)  壱 (U+58F1) | und-Jpan |
| 壹 (U+58F9) | -- | 一 (U+4E00)  弌 (U+5F0C)  壱 (U+58F1) | und-jpan |
| 弌 (U+5F0C) | -- | 一 (U+4E00)  壹 (U+58F9)  壱 (U+58F1) | und-jpan |
| 壱 (U+58F1) | -- | 一 (U+4E00)  壹 (U+58F9)  弌 (U+5F0C) | und-jpan |

In this table, no matter what language tag was set, for any given code point in a variant mapping cluster, its variants must be configured to either “allocatable” or “blocked”, its variants cannot stay as an INDEPENDENT code point, regardless of the fact that in Japanese language environment they are treated as different OLD form and NEW form.

Moreover, only the allocatable code points can be used to generate valid whole label package, and this whole label package will go (be allocated) to the SAME applicant, which means, the co-existence of old form label registrant and new form label registrant will not happen at Root level.