Integration Panel:

Considerations for Designing a Label Generation Ruleset for the Root Zone

REVISION 2014-09-23

This document provides a concise summary of some of the principal issues that a Generation Panel would take into consideration in drafting a Label Generation Ruleset (LGR) for the Root Zone.

Many of these considerations are the same that were followed by the Integration Panel in defining the Maximal Starting Repertoire [MSR]. They are more fully documented in [MSR Overview and Rationale], in particular in Section 7 "Generation Panels' Use of the MSR". Sections 4 and 5 of that document also contain many details that are applicable to the task of developing a script-LGR.

This document is intended as a convenient starting point for Generation Panels as they define the tasks and organize the work of creating a script specific LGR for the Root Zone. It is not intended to supersede the fundamental specification of the tasks as contained in the "Procedure for Developing and Maintaining Label Generation Rules for the Root Zone in Respect of IDN Labels" [Procedure]. In particular, in addition to other requirements, the Procedure sets out a number of process goals and principles that any LGR for the Root Zone is intended to satisfy.

In evaluating a proposed LGR, the Integration Panel will be guided by the [Procedure].

Components of an LGR

Briefly summarized, an LGR for a given script consists of:

1. *Repertoire* – a selected list of single code points, or sequences of code points, that are eligible for use in labels
2. *Code Point Variants* – each element (code points, or sequences of code points) in the repertoire may optionally specify a set of code point variants related to it. Code point variants map a code point, or sequence of code points, onto another code point (or sequence) considered equivalent in some way. For each code point variant should the disposition of a label generated from it should be specified. For details on this, see [Variant Rules].
3. *Whole Label Evaluation Rules* – these may restrict certain combinations of otherwise eligible code points.

See the [Procedure] for the authoritative description of an LGR and its elements.

The following sections in this document contain sets of questions that touch on the main considerations a Generation Panel would take into account in creating a script-specific LGR for the Root Zone. They are grouped according to each element of the LGR. Explanatory notes for some of these are provided at the end.

Answering these questions is intended to assist a Generation Panel in assembling the relevant information needed in making the decisions that lead to the design of the LGR and that should be documented in the rationale document accompanying the proposed LGR. (See also [LGR-Requirements]).

Code point

1. Is it contained in the Maximal Starting Repertoire?
2. Is it used with the script defined in the scope of the GP
    a. do the script IDs match?
    b. is it a combining mark?
    c. is there an alternate representation?
3. Is it suitable in identifiers? 1
    a. is it in widespread modern use?
    b. is it not technical / religious / limited use only?
    c. is it not really a punctuation / symbol?
    d. is it really necessary for representing identifiers?
4. Is the Unicode encoding of the code point stable?
    a. are there any rendering issues?
5. Are there any IDNA2003 compatibility concerns?
6. What are the concerns to DNS security & stability concerns? rendering issue, homoglyph of non-PVALID code points, especially protocol characters (e.g. RFC3986 & RFC3987)
7. How accessible would a TLD containing that code point be?
    a. are there input/keyboard concerns?
8. Are there imminent changes to the writing system that may require "think-ahead" research / design? 9
9. What are the risks if the code point is not included?
10. What are the risks if it is?
11. Is it in tension with any of the Principles in any way? 6
12. Does it always appear in a fixed sequence ? 7

Code Point Variant

1. Would a reasonable person with native knowledge of the script consider a pair of code points interchangeable?
2. Would such a person be unable to determine which code point it is by appearance?
    a. for scripts where appearance varies with position, is this the case for all positions?
3. Does the script have upper/lower case? [10]
4. Is there an alternative representation?
    a. if so, does it involve a Combining Mark? 2
5. Are there anticipated or imminent changes to the writing system that may require "forward-looking" research / design? 9
6. What should the disposition of any defined variants be?
7. Should labels containing the variant be blocked? Otherwise,
    a. should any labels containing the variant be allocatable?
    b. should all labels containing the variant be allocatable?
8. Should any of the variants of this code point be contingent on context?

9. Is there any relationship with code points in other scripts or repertoires? 4
10. Is each set of code point variants symmetric? 5
11. Is each set of code point variants transitive? 6
12. Are any variants contemplated that are in tension with any of the Principles? 3
13. Are the variants designed so that they lead to the minimal required number of allocatable variant labels? 8
14. Are the variants designed so that, in doubtful cases, they block potential variant labels?

Whole Label Evaluation Rule

1. Are there sequences of code points that are only valid in a certain order or fixed sequences? 7
2. Can certain code point only appear in a certain position within a label?
3. Should certain code points be prevented from appearing in a certain position in a label?
4. What is the complexity cost of including a rule?
    a. do related scripts share the same (or a similar) rule?
5. What is the risk of not having such a rule?
    a. what is the risk of having a simplified / less complex version of the rule?
6. Would any defined variants have a different disposition depending on context?
7. Are any rules in tension with any of the Principles?

References

[MSR] Internet Corporation for Assigned Names and Numbers, “Maximal Starting Repertoire”, (Los Angeles, California: ICANN June 2014) https://www.icann.org/news/announcement-2-2014-06-20-en

[MSR Overview and Rationale] Internet Corporation for Assigned Names and Numbers, Integration Panel, “Maximal Starting Repertoire—MSR-1 Overview and Rationale, (Los Angeles, California: ICANN June 2014), https://www.icann.org/en/system/files/files/msr-overview-06jun14-en.pdf

[Procedure] Internet Corporation for Assigned Names and Numbers, "Procedure to Develop and Maintain the Label Generation Rules for the Root Zone in Respect of IDNA Labels." (Los Angeles, California: ICANN, March, 2013) http://www.icann.org/en/resources/idn/variant-tlds/draft-lgr-procedure-20mar13-en.pdf

[LGR-Requirements] Integration Panel, “Requirements for LGR Proposals”, September 2014 https://community.icann.org/download/attachments/43989034/Requirements%20for%20LGR%20Proposals.pdf

[Variant-Rules] Integration Panel, “Variant Rules”, July 2014, https://community.icann.org/download/attachments/43989034/Variant%20Rules.pdf

[XML-LGR] Davies, K. and A. Freytag, "Representing Label Generation Rulesets using XML", http://tools.ietf.org/html/draft-davies-idntables/.Visited 2014-07-06.

Notes

1. The Maximal Starting Repertoire [MSR] represents a rough cut with respect to identifier eligibility. In further "short listing" the code points for the actual repertoire, the Generation Panel is expected to establish the actual suitability for each code point.
2. IDN labels are required to be in Unicode Normalization Form C (NFC). Normally, that means it is not possible for a label to contain a sequence of base character and Combining Mark if a precomposed form exists. However, some combinations of base character and combining mark can be used to create the appearance of an existing single character, even though the latter is unrelated and has no formal decomposition in Unicode.
   An appropriate response in such a case might be to make the single character and the sequence *blocking variants* of each other or to omit the Combining Mark from the repertoire.
3. Whenever a proposed code point, variant mapping or WLE is in tension with one or more of the Principles, the Integration Panel would expect a particularly strong rationale for their inclusion in the LGR. That said, the principles themselves are in some tension, and in tension with the overall process goals laid out in the [Procedure].
4. Blocked variants may exist across repertoire boundaries. Specifying them may be appropriate where homoglyphs exist across repertoire boundaries. Possible examples include Han ideographs and Latin/Greek/Cyrillic homoglyphs. (Homoglyphs are identical in appearance, but have intentionally been given separate code points).
5. An LGR is symmetric if the following is true for all code points A and B: if A → B, the LGR also specifies B → A, where A → B means B is a variant of A. (See also [Variant Rules]).
6. An LGR is transitive if the following is true for all code points A, B and C: if A → B and B→ C, the LGR also specifies A → C), where A → B means B is a variant of A. (See also [Variant Rules]).
7. For code points that occur only in a small number of fixed sequences it may be appropriate to limit their occurrence to these sequences. In some cases this can be done by making the sequence itself a repertoire element. This does not require an actual WLE Rule. See [XML-LGR]. An example are digraphs of Hebrew code points used in Yiddish.
8. See [Variant Rules].
9. Examples might be a new code point that would be a variant of an existing code point, but that is currently not encoded, but pending for addition to a future version of Unicode, or an anticipated change in the status of a code point making it eligible to be added to the MSR or LGR in the future (expected to enter widespread use, perhaps as result of a pending reform of a supported orthography).
10. Only lowercase is allowed in IDNA 2008 proper. However, browsers accept uppercase names and map them to lowercase before resolving. To give an example, it means, that to a user the sequence of three Greek capital alphas (AAA) will resolve to same label as ααα where α is the lowercase alpha. While ααα will be distinct from 'aaa', where 'a' is the Latin letter, Greek alpha (A) and Latin A are homoglyphs. In this example, the question is whether it is acceptable to allow both a label aaa and a label ααα to be registered in the root, given that users thinking of either label as AAA cannot predict which label this will resolve to.