Label Generation Rulesets
Next generation “IDN tables” in XML
ICANN 50, London, UK

Kim Davies
Director, Technical Services
The quick summary

- A standardised approach to expressing registration rules for labels, come to be known as “label generation rulesets”, or “LGRs”.
- Should allow all existing IDN tables, and known registry policies, to be reproduced in an objective machine readable format.
- Vision is anyone dealing with registry policies can implement an LGR runtime, and then not have to worry about hardcoding complex validation rules.
- Is not specific to IDNs!
Why? (Part 1)

- Current IDN implementors tend to use “IDN tables” to define which code points are allowed in domain labels.
- Many registries implementing contextual logic in their registry backends in a bespoke manner.
- Therefore, even for those that publish IDN tables, it is difficult to be sure you can replicate/reuse that logic.
ICANN maintains an “IDN repository” of registry IDN tables.

Lacks consistent format and difficult to repurpose registry tables.

RFC 3743 and RFC 4290 are not rich enough to express most registry policies.

New gTLD Program has struggled with this too.

Having a common format would greatly aid in re-use, validation for table format, etc.
Why? (Part 3)

- IDN variant program needs to use something to base its work on.

- One goal is to have a master “root LGR” which is the unison of various language and script specific rulesets.

- Therefore the ability re-use, adapt, merge from LGRs is a requirement.
draft-davies-idntables

- XML based description of registry policies for “label generation”
- Allowable code points for domain registries, contextual rules, dispositions, etc.
- Co-authored by me and Asmus Freytag (Unicode Consortium); strong input from IDN variant project participants
Simple validation checking

Provide label, respond whether that label accords with the LGR’s rules.
**Variant label generation**

Take an input label, generate permutations along with actions to take.
LGRs can be merged
LGRs can be diffed
LGRs can represent complex interdependencies
LGRs have...

- Code point lists, with tagging classes and dispositions
- Variants for specific codepoints, variants are 0..n codepoints. Variants can be conditional by meeting certain tests.
- Whole label variants, allowing rules based on regex like concepts.
- Leverages all the Unicode properties, to diminish the need to be derivative.
- Metadata in standard format
- A clear schema so table validity can be automatically checked.
<?xml version="1.0" encoding="utf-8"?>
<lgr xmlns="http://www.iana.org/lgr/0.1">
  <data>
    <char cp="002D" comment="HYPHEN (-)" />
    <range first-cp="0030" last-cp="0039" comment="0-9" />
    <range first-cp="0061" last-cp="007A" comment="Latin small letter A-Z" />
  </data>
</lgr>

HelloWorld.lgr
Minimal LGR to permit standard LDH labels
These rule allows zero-width joiner (U+200D) when following a virama (implements IDNA context rule rule)
Current status

- Is the standard format for the IDN variant project.
  - Community “generation panels” are developing their language/script rules using the format.
  - “Integration panel” will merge these inputs into a common Root LGR.
  - As a starting point, Integration panel created a “Maximal Starting Repertoire” (MSR) using the format. i.e. https://www.icann.org/en/system/files/files/msr-wle-rules-06jun14-en.xml
  - Folks that have reviewed it tell us it is a good idea.
  - No-one has said it is a foolish idea (yet.)
Long term vision

- Make this the standard format for the IDN repository at iana.org.
  - All existing tables can be ported to the new format
  - We can then automate the IDN table repository, do automatic validation checking, and other neat stuff.
- Make this the required format for the next round of gTLD applications.
  - Will address many issues seen with the current approach.
- Registries will typically implement an LGR engine in their backend, and slot in LGRs based on business requirements.
What’s next?

• Specification is stabilising, we think it is getting mature. Work swings to completing software implementations.

• Don’t intend to finalise until we have real-word experience in production (variant work)

• We think it has broader utility than just variant work, would benefit from wider review.

• A strong ecosystem of implementations and uses would benefit all, and reduce costs. Less need to reinvent the wheel.

• Open question whether to move it into the Standards Track within the IETF.
Thank you.

- kim.davies@icann.org