Designing Secure and Stable Rules for Internationalized Domain Names

Pitinan Kooarmornpatana
IDN Program, Senior Manager

ICANN DNS Symposium 2023
5 September 2023
Agenda

- Internationalized Domain Names for Applications (IDNA) 2008
- Label Generation Rules
- Designing Secure and Stable IDNs Under Your TLD
Internationalized Domain Names for Applications (IDNA) 2008
IDNA 2008

- Internationalized Domain Names for Applications (IDNA) is a set of standards for handling Internationalized Domain Names (IDNs).
- Developed by the Internet Engineering Task Force (IETF).
- The current version is IDNA 2008, consisting of:
  
  **Standard Track:**
  
  - RFC 5890 IDNA: Definitions and Document Framework
  - RFC 5891 IDNA: Protocol
  - RFC 5892 The Unicode Code Points and IDNA
  - RFC 5893 Right-to-Left Scripts for IDNA

  **Informational Track:**
  
  - RFC 5894 Background, Explanation, and Rationale
  - RFC 5895 Mapping Characters for IDNs in Applications (IDNA) 2008

- All IDNs must be IDNA 2008 compliant.
Example: Latin Script

- **RFC 5892**, Introduction section states: “**PROTOCOL VALID [PVALID]**: Those that are allowed to be used in IDNs. Code points with this property value are permitted for general use in IDNs.”

- Latin Extended-A code point range: **0100–017F**.

- Based on the [IDNA2008 Derived Property Values](https://datatracker.ietf.org/doc/html/rfc5892#appendix-B), only code points with PVALID property can be considered for IDNs.

<table>
<thead>
<tr>
<th>Code Point</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100</td>
<td>DISALLOWED</td>
</tr>
<tr>
<td>0101</td>
<td>PVALID</td>
</tr>
<tr>
<td>0102</td>
<td>DISALLOWED</td>
</tr>
<tr>
<td>0103</td>
<td>PVALID</td>
</tr>
<tr>
<td>0104</td>
<td>DISALLOWED</td>
</tr>
<tr>
<td>0105</td>
<td>PVALID</td>
</tr>
<tr>
<td>0106</td>
<td>DISALLOWED</td>
</tr>
<tr>
<td>0107</td>
<td>PVALID</td>
</tr>
</tbody>
</table>

Example: Latin Script

- **0100 DISALLOWED**: LATIN CAPITAL LETTER A WITH MACRON
- **0101 PVALID**: LATIN SMALL LETTER A WITH MACRON
- **0102 DISALLOWED**: LATIN CAPITAL LETTER A WITH BREVE
- **0103 PVALID**: LATIN SMALL LETTER A WITH BREVE
- **0104 DISALLOWED**: LATIN CAPITAL LETTER A WITH OGONEK
- **0105 PVALID**: LATIN SMALL LETTER A WITH OGONEK
- **0106 DISALLOWED**: LATIN CAPITAL LETTER C WITH ACUTE
- **0107 PVALID**: LATIN SMALL LETTER C WITH ACUTE
Additional Requirements in IDNA2008

◉ RFC 5892, Introduction section states:
  “… However, that a label consists only of code points that have this property value [PVALID] does not imply that the label can be used in DNS. See the Protocol document [RFC5981] for algorithms to make decisions about labels in domain names.”

◉ RFC 5890, Section 2.3.2.3 states:
  “Because of the diversity of characters that can be used in a U-label and the confusion they might cause, such restrictions [“variant definitions and rules beyond those imposed by DNS or IDNA”] are mandatory [emphasis added] for IDN registries and zones even though the particular restrictions are not part of these specifications (the issue is discussed in more detail in Section 4.3 of the Protocol document [RFC5891]).”

◉ Additional restrictions beyond IDNA are mandatory, therefore, the Label Generation Rules (LGRs) are needed.
Label Generation Rules
IDN Tables as Label Generation Rules

- IDN Implementation Guidelines
  
  **Version 3.0** states: “All such code point listings will be placed in the IANA Repository for IDN TLD Practices in tabular format together with any rules applied to the registration of names containing those code points, before any such registration may be accepted.”

  **Version 4.1** states: “The IDN Table must include the complete repertoire of code points, any IDN variant code points and any applicable contextual rules which the TLD registry uses to determine if an IDN label is acceptable for registration.”

- For a TLD registry operator to offer IDNs at the second level, the rules governing which IDN labels are allowed for registration must be defined to ensure the secure and stable solution for the Domain Name System.

- Each language or script required a set of rules. The ruleset is called IDN Table or Label Generation Rules.
Consistency with the Root Zone LGRs

- The Root Zone Label Generation Rules (RZ-LGR) are developed by various script communities to ensure secure and stable IDN TLDs.

- RZ-LGR Version 5 is available for 26 scripts covering more than 386 languages.

- SAC060 recommended “ICANN should coordinate and encourage adoption of these rules [RZ-LGR] at the second and higher levels…”
Label Generation Rules

- Consist of three main aspects:
  - Repertoire
  - Variants
  - Rules
What is the Goal?

- Goal is to create a mnemonic system for use in the Domain Name System (DNS).
  - A mechanism to remember IP address.
  - Must remain secure and stable in use – if the DNS is confusing to users, then the motivation is not met.
  - Not required to completely cover a language or a script.
  - May not form labels that are words in a language.
    - Not restricted to “correct” spellings.
    - May not carry a meaning in the “lexical” sense.
Internationalized Domain Name (IDN) Labels

Syntax of IDN Labels
Valid U-Label: Unicode code points as constrained by the “LDH” scheme within IDNA 2008

Syntax of IDN Labels
Valid U-label, further constrained by the “letter” principle for TLDs

gâteau.café
### IDN Mnemonics

#### C1 Controls and Latin-1 Supplement Code Chart

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>35</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
<td>69</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>01</td>
<td>02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>81</td>
<td>82</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>91</td>
<td>92</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>A1</td>
<td>A2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B0</td>
<td>B1</td>
<td>B2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C0</td>
<td>C1</td>
<td>C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>D0</td>
<td>D1</td>
<td>D2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E0</td>
<td>E1</td>
<td>E2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F0</td>
<td>F1</td>
<td>F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Note:** The image contains a grid of characters with corresponding hexadecimal codes.
Designing Repertoire

- Starting from **RFC6912** - Principles for Unicode Code Point Inclusion in Labels in the DNS:
  - **Longevity** – stable across Unicode versions
  - **Least Astonishment** – take into account the population using a code point
  - **Contextual Safety** – sensitive to ways in which code point may be used in malicious ways
  - **Conservatism** – any code point inclusion decision is as conservative as practicable
  - **Inclusion** – default is excluded, then add code point which is safe based on usability and confusability
  - **Simplicity** – rules determining use should be simple to understand
  - **Predictability** – rules determining whether a code point is included are predictable for others to reach the same conclusion
  - **Stability** – if a code point is permitted, taking it out is very hard
Label Generation Rules

- Consist of three main aspects:
  - Repertoire
  - Variants
  - Rules
What is the Goal?

- Successfully defining variant rules for an LGR is not trivial.
- Code point or code point sequences causing two (or more) labels functionally “the same” in a script.
- Use the mnemonic system to minimize user confusion.
- Conservatism requires:
  - Maximizing “blocked” variants
  - Minimizing “allocatable” variants
Examples of Variants for Latin Script as per RZ-LGR

◉ Visual Variant Pairs
  U+011F ģ Latin Small Letter G with Breve and
  U+01E7 ģ Latin Small Letter G with Caron

  U+0169 ũ Latin Small Letter U with Tilde and
  U+016B ū Latin Small Letter U with Macron

◉ Non-Visual Variant Pairs
  U+0066 f Latin Small Letter F and
  U+0192 ƒ Latin Small Letter F with Hook

  U+0073 U+0073 ss Latin Small Letter S + Latin Small Letter S and
  U+00DF ß Latin Small Letter Sharp S

◉ For complete variant sets please see the Latin Script RZ-LGR Proposal.
Label Generation Rules

- Consist of three main aspects:
  - Repertoire
  - Variants
  - Rules
What is the Goal?

- Goal is to reduce label space. Preventing labels that should not be possible for various reasons:
  - Not licensed by the script (but not related to spelling rules) or cause security issues.
  - Cause usability constraints.
- Reducing allocatable label by making them blocked in certain cases.
Examples

- Based on IDNA2008 ([RFC5892](https://tools.ietf.org/html/rfc5892))
  - U+00B7 ⋅ MIDDLE DOT must be between 'l' (U+006C) characters only, used to permit the Catalan character ela geminada to be expressed.

- Based on the [Latin script RZ-LGR](https://www.unicode.org/reports/tr29/)
  - U+0304 ⸄ Combining Macron is not included as a stand alone code point, but always in a sequence:
    - 006E 0304 ň Latin Small Letter N with Combining Macron, used in Raga (Hano) and Marshallese languages
Example

Based on the Thai language reference LGR

The following 2 code points are different vowels:

- U+0E40 เ THAI CHARACTER SARA E
- U+0E41 แ THAI CHARACTER SARA AE

Defining a rule to disallow 0E40 to follow itself could reduce the homoglyph confusion with 0E41

แมว (cat)

แมว (cat)

แมว (cat)
Designing for Secure and Stable IDNs Under Your TLD
Reference Label Generation Rules (LGR)

- **Background**
  - Based on script user community input for 26 scripts garnered through the RZ-LGR project. ICANN org developed second level reference LGRs derived from RZ-LGR with some additions to remain consistent.
  - All reference LGRs include consultations with the script communities and are finalized after Public Comment proceedings.
  - TLD registry operators can use reference LGRs when they develop rules for IDN registration under their TLDs.
  - The online LGR Tool is available for reviewing the designed IDN table with a reference LGR.
  - ICANN org uses reference LGRs and the LGR Tool to review the registration rules under gTLDs.
  - This contributes to consistency and transparency of the review as requested by the Registries Stakeholder Group (RySG).
Reference Label Generation Rules (LGR)

- Additional 7 script-based LGRs and related updates will be published which will bring the number of reference LGRs to 54.

- 31 language-based LGRs
  Arabic, Belarusian, Bosnian (Cyrillic), Bosnian (Latin), Bulgarian, Chinese, Danish, English, Finnish, French, German, Hebrew, Hindi, Hungarian, Icelandic, Italian, Japanese, Korean, Latvian, Lithuanian, Macedonian, Montenegrin, Norwegian, Polish, Portuguese, Russian, Serbian, Spanish, Swedish, Thai, and Ukrainian.

- 23 script-based LGRs
  Arabic, Armenian, Bangla (Bengali), Cyrillic, Devanagari, Ethiopic, Georgian, Greek, Gujarati, Gurmukhi, Hebrew, Japanese (Hiragana, Katakana, Kanji[Han]), Kannada, Korean (Hangul, Hanja [Han]), Khmer, Lao, Latin, Malayalam, Myanmar, Oriya, Sinhala, Tamil, and Telugu.
Collaboration on Developing Reference LGRs

- If any additional reference LGR is needed for your target users, TLD registries are encouraged to cooperate and contribute toward the development of and updates to the Reference Second Level LGRs.

- Script user communities, TLD registries, and ICANN org will collaborate on developing the additional reference LGRs.

- Please contact us at IDNProgram@icann.org
Engage with ICANN – Thank You and Questions

One World, One Internet

Visit us at icann.org

@icann
facebook.com/icannorg
youtube.com/icannnews
flickr.com/icann
linkedin/company/icann
soundcloud/icann
instagram.com/icannorg