Reference Label Generation Rules (LGR) for the Second Level — Overview and Summary

REVISION – January 11, 2022

Table of Contents

1 Overview

This document describes a set of Reference Label Generation Rules (LGRs) for the Second Level. These reference LGRs were developed according to the "Guidelines for Developing Reference LGRs for the Second Level" [Guidelines]. For a particular subset of language-based LGRs, the guidelines define a process that builds on the results of a previous project [IIS] but provides additional review and development, documentation and translation to XML [RFC7940]. In some cases, these LGRs extend the repertoire compared to [IIS]. All other LGRs are derived from the Root Zone LGR for the corresponding script, with additional input from the same panels that developed the Root Zone LGRs. The reader of this document is assumed to be familiar with the [Guidelines].

This document provides some general background related to the design and design process for these LGRs, as well as general considerations relevant to anyone wishing to adopt or adapt these LGRs. Not all reference LGRs are developed on the same schedule, this document applies to all that have been published or are actively under development. Some statements may not apply to every LGR.

The LGRs are specific to a given language or script (and in some cases the combination of language and script) but not necessarily specific to a geographically compact user community. Each file has been reviewed by the community and some have seen additional review by one or more linguistic experts, as well as reviewed by a separate expert for DNS stability and security issues.

1.1 Reference Label Generation Rules (LGR) Files

The normative definition of each reference LGR is provided as an XML file. The LGRs are expressed using a standard format defined in "Representing Label Generation Rulesets in XML" [RFC7940].

Each of these files contains all the LGRs applicable to the labels from that language, and only those rules. Each file contains a complete description, a repertoire with optional variants, and WLE Rules, as well as detailed references that link each included code point to a reference providing data for justifying its inclusion.

From each XML file, a non-normative HTML presentation is generated mechanically. These are provided for the convenience of the reader. The HTML presentation is augmented by summary data as well as data extracted from the Unicode Character Database [UCD].

The set of [Second-Level Reference] files can be found on this website:
https://www.icann.org/resources/pages/second-level-lgr-2015-06-21-en

2 Notes

The development and review proceeds according to the [Guidelines]. The following notes provide some additional highlights as well as information not specific to any individual reference LGR.

2.1 Repertoire

The repertoire for each Reference LGR is either based on the Root Zone LGR with suitable additions for the second level, or based on a consensus repertoire derived from the sources consulted. In the case of many of the language-based LGRs, this repertoire caters to more than the code points minimally needed to write the native vocabulary of the language by including code points that are in common use for loan words and the like. Where a language has multiple user communities with some variation of usage, a single, combined LGR is produced. The details are described in each of the LGRs.

2.1.1 Root Zone LGR and Second Level Additions

Many of the reference LGRs are directly derived from the relevant Root Zone LGRs. These LGRs deliberately limit their repertoire to code points in everyday common use, eschewing historic, obsolete or special purpose code points in limited use.

In some cases, the repertoire has been further limited to code points used in a specific language, by using language-specific source information provided in that LGR.

In all cases there have been additions to the repertoire specific to the second level. Generally, the HYPHEN-MINUS has been added and either the common (ASCII) digits or the set of script-specific digits for that script or both. In isolated cases, some commonly used characters ineligible for the Root Zone were added as well. For some scripts, CONTEXTJ code points are used in an integral way in spelling many words and it may be both possible and desirable to support them with strict limits and proper safeguards in a second level LGR. All such additions generally proceed in consultation with experts from the same panels that developed the Root Zone LGRs.

2.1.2 Sources for Repertoire other than the Root Zone

In determining the repertoire where it is not derived from the Root Zone LGR, a large number of sources was investigated, from spelling dictionaries released by language authorities, RCFs and national or international standards, to other sources such as ordinary dictionaries, the Common Locale Data Repository (a project of the Unicode Consortium) [CLDR] and finally existing IDN practice for ccTLDs aimed at users of a native language. The sources and their contribution to the development of the repertoire are documented in detail in each of the LGRs.

In some cases there has been direct community input into the design of the Reference LGR.

2.2 Extended Code Points

Many, though not all, of the languages are written by compact communities that are in contact with other languages in the same region or in the same country. In those cases, native users may have familiarity with or need for access to an extended set of code points, for example for names of people or places. Certain of the Reference LGRs languages provide for those code points by listing them as “extended-cp” if they are not already catered for in the core repertoire. As written, the LGRs treat these extended code points as ineligible for a label, but registries could easily remove the restriction to tailor such reference LGR to their needs. (See also Section 2.6.2, “Reference LGR-specific rules”).

This is in contrast to script-based LGRs that typically provide for the full repertoire needed for all languages sharing a common script, or those country-based LGRs, that provide for the needs of users from the same country or territory, irrespective of whether they write a majority or minority language prevalent in the country.

2.3 Excluded Code Points

For most languages or scripts, there are among the consulted sources some that include a number of code points that are very rarely used, or that are historic or limited to special purposes, like poetry and religious works. Such uses are rarely germane to IDNs. Consequently, such code points have been excluded from these reference LGRs and documented as such; either explicitly in the LGR, or implicitly in the background documentation for the cited Root Zone LGR development (see, for example, code points excluded from the [MSR]). In contrast to the “extended” code points, which are specifically called out as likely targets for customization, excluded code points can be safely ignored. Where it is felt that some might be required, the reason could well be that the scope of the LGR no longer fits the original design, but has morphed from, for example, a language-based LGR to a regional or multilingual LGR. That would be an indication that the chosen reference LGR doesn’t really apply and a better approach might be to cut down a script LGR to size.

2.4 Sequences and Context Constraints on Repertoire Elements

Certain code points tend to occur only in fixed combinations. The repertoire contains such code points only as part of an explicitly specified code point sequence. This prevents unneeded combinations and primarily applies to combining marks such as diacritics (but also to certain clusters in complex scripts). Rendering systems may fail to provide a predictable presentation of combining marks if they are present outside of expected contexts, whether applied to unexpected base characters or, for example, repeatedly applied. In the latter case, in particular, there is a danger of “overprinting”, which would mask the presence of the extraneous diacritic. Finally, some diacritics are easily confused with one another. Allowing unrestricted combinations would allow these diacritics to be applied to base characters that normally take different diacritics, greatly adding to the risk of creating confusable labels. Additional constraints are provided by Whole Label Evaluation and Context Rules (see Section 2.6).

2.5 Variants

Several of the language-based LGRs do not include the definition of any variants, but most of the script-based LGRs do. Where variants are included, their selection is informed by existing registry practice, as well as by the work performed at ICANN on the script LGRs for the Root Zone. Any cross-repertoire (or cross-script) variants identified in the Root Zone have been retained here for use in zones that support reference LGRs for more than one script or language. (See Section 4, “Use of Multiple Reference LGRs in the Same Zone”).

2.5.1 Digit Variants

All the LGRs support the common (ASCII) digits. Any script-specific (or native) digits are treated as semantic variants of the corresponding common digits. In zones where multiple scripts are present, all digit sets would become semantic variants of each other as required by transitivity.

In a few cases, different sets of native digits across scripts share forms that suggest the need to include variants based on homoglyph relations. There are cases where digits of different numerical value are homoglyphs of each other and defining them as cross-script variants would create potential conflict with variants based on numeric equivalence.

Some scripts share a single shape for use as letter or digit, as a consequence of transitivity, these letters will become cross-script variants of all corresponding digits in the other scripts (by value). However, the common digit 0 is not treated as a variant of letter ‘o’ – because that variant relation does not exist in LDH (non-IDN) labels.

If support were added later for the native digits for any script, some in-script variant relations might need to be added. In zones where multiple scripts are present, it might be difficult to accommodate both the required in-script homoglyph variants as well as variants based on numeric equivalence; thus extensions to the sets of supported digits need to proceed with extreme caution and deliberation.

Finally, some native digits can be understood as having a potential homoglyph relation to letters outside the script. It is generally not possible to satisfy both the semantic variant relation and such cross-script homoglyph variants in a consistent way, while maintaining transitivity. As a result, these reference LGRs prioritize the semantic relationships for digits, with the thought that not all second level LGRs are necessarily in zones that support multiple scripts, but for which allowing two sets of digits immediately creates an in-script issue. Therefore, such issues would need to be addressed outside the reference LGR, for example via additional registry policies.

2.5.2 Variant Dispositions

All LGRs with variants support the disposition “blocked”, meaning that either a label or its variant may be allocated, but never both. Some LGRs support “allocatable” variants, meaning that both the label or the variant or both may be allocated to the same registrant. A few LGRs contain additional dispositions, such as “activated”, which implies that a label should be allocated. There are “optional” forms of some variant types defined that allow for simple customization of variant dispositions as described in the particular LGR. (In all LGRs, the original label, if not blocked or invalid, is reported as “valid”.)

2.5.3 Multiple allocatable variants

The Chinese LGR resolves variant labels into “allocatable” and “blocked” only, however it utilizes a rather specific form of variant subtyping in an attempt to keep the number of allocatable variants manageable. The details are described in the LGR and references cited therein.

Other LGRs use whole label evaluation rules to limit the available variant labels to those that consistently use either the one or the other variant code point throughout the label, disallowing mixed labels. In some cases, this is implemented by “no-mix” rules for the specific variants, in other cases there are rules that require that all variants are in a single sub-repertoire (for example, a repertoire for a specific language). Occasionally, both of these approaches are used.

Because the DNS does not natively support variant labels there is a cost to having multiple variants delegated, and thus a need to add such mitigation. The mitigation approaches in a particular reference

LGR are not always sufficient on their own because it may not be possible to write generalized rules preferring some variants to others. Thus, registries should implement additional policies to limit the number of variant labels actually delegated.

Please note that even where typical labels may not generate an inordinate number of labels, for some LGRs certain pathological labels may be created for which the theoretical number of allocatable variants could cause enumeration of allocatable variants to fail by exhausting resources.

2.5.4 Multiple Blocked Variants

There is no limit to the number of blocked variants a label may produce under any of these Reference LGRs. Testing for collision between labels that are variants of each other should be performed by computing a single “index variant” followed by a comparison of these index variants. The performance of this operation does not depend on the theoretical number of blocked variants (For a more detailed description, and for additional notes how to ensure that index variant calculation is well-behaved, see [RZ-LGR-Overview], but also see Section 4 below).

2.5.5 Context rules for Variants

Some variants require context rules to be well-behaved. See RFC 8228 or the discussion of this issue in Section 6, “Design Notes for the Root Zone LGR” in [RZ-LGR-Overview]. Any such context rules from the corresponding Root Zone LGRs have been retained, and a few additional ones have been added where needed by new repertoire (for example, see the Devanagari LGR’s treatment of the HYPHEN).

2.6 Whole Label Evaluation (WLE) and Context Rules

WLE and context rules implement a further constraint on labels, for example, by limiting certain code points from occurring at the beginning of a label, repeatedly, or simultaneously with other code points in the same label. Context rules are a form of a WLE rule that defines a constraint on the surrounding context for a given code point at that position in a specific label (see [RFC7940]).

2.6.1 Protocol-defined Rules

Because the XML format for the LGR supports machine-evaluation of labels for validity, these reference LGRs include all relevant constraints on labels defined in the IDNA protocol itself. In this way, the LGRs can be used to validate all constraints on the label in one pass. For the purpose of these reference LGRs, an additional rule preventing the mixing of any two digit sets in the same LGR has been adopted project-wide.

Common rules:

- Hyphen Restrictions — restricts the allowable placement of U+002D (-) HYPHEN (no leading/ending hyphen and no hyphen in 3-4 position). These constraints are described in section 4.2.3.1 of [RFC5891].
- Leading Combining Marks — restricts the allowable placement for combining marks (no leading combining mark). This constraint is described in section 4.2.3.2 of [RFC5891].
- Digit Mixing — all LGRs support a rule to prevent mixing of multiple sets of digits in the same label.

Rules for Right-to-Left labels:

- Leading Digit — restricts the allowable placement of digits for right-to-left labels (no leading digit in RTL label). This constraint is described in section 2.1 of [RFC5893].
- Mixed Digits — prevents the mixing of European and Arabic (Indic) digits. This constraint is described in appendix A.8 and A.9 of [RFC58932]. (In these reference LGRs this is a subset of the general digit mixing restriction.)

Context rules:

- Japanese in Label — restricts the occurrence of KATAKANA MIDDLE DOT to labels containing at least one code point from any of these scripts: Han, Hiragana, or Katakana; This rule is described in Appendix A.7 of [RFC5892].

Whole label rules:

- No ASCII only Label —restricts IDN labels to those having at least one non-ASCII code point. {RFC 5891]. [1]

For these reference LGRs, the protocol-derived rules, other than the common rules, have only been included if they are needed for labels in the given script. The description section of each LGR file lists the rules and their associated references.

If a label to be validated has already been tested against protocol-derived constraints by the time the LGR is applied, these rules would be redundant and could be removed.

2.6.2 Reference LGR-specific rules

A small number of the LGRs contain additional LGR-specific WLE rules, reflecting a further constraints on possible labels based on the nature of the language or script. These are documented in detail in the description section of the respective LGRs. These rules are generally derived from the Root Zone LGRs with suitable extensions in case of added repertoire elements (for example, see the Thai language LGR for the addition of a Thai abbreviation mark with Thai-specific rules).

The majority of these rules take the form of context rules on a given code point or sequence.

In the case of many complex scripts, readers and layout engines expect the label to be series of valid syllables according to the rules of the writing system. Even though Unicode may encode the individual components of such syllables, arbitrary sequences of code points are not only unexpected, but can lead to security and predictability issues for identifiers. This is in contrast to alphabetic or ideographic scripts where the letter or ideograph is considered the unit, and arbitrary sequences are being used in identifiers. For that reason, complex script LGRs have WLE or context rules enforcing the deep syllabic structure of the writing system, while generally not attempting to enforce “spelling rules”. While it may

[1] By community request, this rule is not enforced in these reference LGRs, meaning that they can be used to apply for whatever subset of LDH labels the LGR may permit (most commonly ASCII digits plus Hyphen).

thus not be possible to make labels using non-existing syllables, there are generally no restrictions against creating "nonsense" words from well-formed syllables.

2.6.3 Special rule for optional repertoire items

Finally, some of the second level reference LGRs use a special context rule to support adapting a reference LGR to a specific zone. (For details, see Section 2.2). By default, this rule may be present in any table, whether actually associated with any code points or not.

Special rule present in some reference LGRs:

- Extended-cp —this context rule always fails. That means, as published, the LGRs do <u>not</u> allow the code points identified as extended by having been given a context of "extended-cp".

Simply changing that rule so it always matches would enable the entire set of extended code points without the need to edit the list of characters. Alternatively, the context condition could be removed from individual code points, thus enabling them one by one. Likewise, to create a subset, a code point can be disabled by adding the "extended-cp" context condition. Doing so would mark the code point as deliberately not included instead of merely omitted.

Alternatively, a copy of that rule, named "excluded-cp" could be used for the latter purpose.

2.7 Metadata

The XML file format defines a number of elements for metadata. Several elements are not relevant to reference LGRs, but would be relevant to actual, deployed LGRs. These elements include <scope>, <validity-start>, and <validity-end>.

In adopting a reference LGR as the LGR for a specific zone, values for these elements should be supplied. For more details, see [RFC7940].

3 Use of Reference LGRs

The information in these reference LGRs represents the best available knowledge of the code points suitable for IDNs for users of a given language or script. As [RFC 6912] makes clear, IDNs are intended to be *reasonable mnemonics*, not text that is in a given language. However, what is a reasonable mnemonic is informed by the language of the user community. Letters or diacritics that are unfamiliar in appearance do not make good mnemonics. In that sense, the fact that these LGRs have been developed for a given language can also be understood as meaning that they were developed with users of a particular language in mind.

3.1 Subset Repertoires

Creating a subset of one of these LGRs would generally represent a more conservative choice (see [RFC6912]). However, the final choice will always have to be made in tension between the two goals of usability and conservatism. There are several issues to consider when contemplating the creation of a subset. The first affects usability. For example, consider the case of reducing any Latin-based LGR to the letters "a-z". This is undoubtedly a conservative choice. But, it also eliminates any gain in usability compared to non-IDN labels. A subset should always be a carefully designed consistent whole. (See also Section 2.2 and Section 2.6)

The next concern applies to LGRs that contain variants. For those LGRs, the effect of subsetting on the variant sets must be reviewed thoroughly. For each code point to be removed, all variant mappings related to that code point must also be removed. Once these are removed, it may not be possible to add the code point back again in a future version of the LGR due to the risk of stability issues. Finally, any rule that depends on the definition of a given code point must be updated if that code point is removed.

3.2 Overlapping Repertoires

Additional policies, variants and rules may be needed if any of these reference LGRs is adopted along with other LGRs that have an overlapping repertoire. This is especially relevant in the case of LGRs defining variants for LGR-specific rules.

3.3 Repertoire Extensions

There are two ways of extending these LGRs. The first is by allowing additional code points that are considered widely used in the context of a given language or script. Some of the reference LGRs directly provide information on a suitable set of such extended code points (see Section 2.2).

The second is the use of a number of language-based reference LGRs as “building blocks” in assembling local, regional or script-based LGRs. When used in that fashion, care must be taken so that the resulting LGR provides for a consistent treatment of variants and WLE rules.

Any combined LGRs should be from a common script. The issue of mixed script labels is addressed in [RFC5890]. In combining LGRs into a single LGR it is recommended to first combine their core repertoires and, after eliminating duplicates, to consider possible additions from the extended sets separately. A combined LGR could have multiple “language” elements to indicate the range of languages covered, or a single language element indicating the script (see [RFC7940]).

3.4 Variants and Rules

When merging LGRs, or when using LGRs with overlapping repertoires in the same zone, the “rules” element in the XML must be given special scrutiny. Some of the “rule” and “class” elements may be merged safely. Others may have to be renamed to keep them distinct. “Action” elements must be present in the order required for proper precedence in the merged XML.

That said, most of the LGRs presented here have a generic, default “rules” element. Any two LGRs with only the default rules can be merged and a single copy of the default rules appended.

4 Use of Multiple Reference LGRs in the Same Zone

If a zone supports multiple reference LGRs, cross-repertoire (or cross-script) variant labels may exist (see Section 6.3 “Cross-script variants” in [RZ-LGR-Overview] for a discussion). This situation arises when multiple LGRs are used, each defining the valid labels and variants for a given script or language (in contrast to the case of a combined repertoire as discussed in Section 3.3, “Repertoire Extensions”). For efficient resolution of cross-repertoire variants, a special merged or “common” LGR needs to be created that is optimized for the task. For a discussion, see Section 5.2 “Common LGR” in [RZ-LGR-Overview]). As long as the common LGR file was created using all of the multiple LGRs that have cross-repertoire variants with each other, it can be used for that purpose – even if it contains information from additional LGRs.

Note that these reference LGRs are designed with the assumption that any native digits are variants of the corresponding ASCII digits, if both occur in the same LGR. When multiple LGRs are used in the same zone, transitivity would require that all such native digits are cross-script variants of each other. The effect of this transitivity is omitted from the individual LGRs; it would have to be applied to a common LGR as part of the merge process to enable for proper cross-repertoire collision detection.

Common LGR files for some subset of these Reference LGRs may be provided. If so, the preamble to the file states which files were included in its preparation.

5 Review

These reference LGRs have been reviewed by the community as part of a public review process. Certain of the reference LGRs were additionally reviewed by members of the corresponding Root Zone Generation Panels or by independent reviewers with expertise in Unicode and linguistics, as well as IDNA and DNS security. Any LGRs were updated to reflect the input from the review. For the independent expert reviewers, review results where available are found at [Second-Level Reference].

5.1 Versioning

Each reference LGR has a version number and a date of publication. Any changes or corrections to a published reference LGR result in a new version number and publication date. The nature of any corrections or changes is documented in the description.

6 Contributors

The following cumulatively lists contributors to the development of any Second Level Reference LGRs.

1. Developers

Asmus Freytag
Michel Suignard

2. Expert Reviewers

Michael Everson
Nicholas Ostler
Lu Qin
Wil Tan

3. ICANN Staff

Sarmad Hussain
Pitinan Kooarmonpatana
Anand Mishra

4. Root Zone Generation Panel Experts

An early version of all script-specific and some language-specific LGRs was reviewed by experts from the Root Zone LGR project. For a list of members for each Generation Panel, see the proposal document for the corresponding Root Zone LGR.

7 References

[CLDR] CLDR - Unicode Common Locale Data Repository: http://cldr.unicode.org

[Guidelines] Internet Corporation for Assigned Names and Numbers, "Guidelines for Developing Reference LGRs for the Second Level". (Los Angeles, California: ICANN, October 2015) https://www.icann.org/en/system/files/files/lgr-guidelines-second-level-27may20-en.pdf

[IIS] IIS, IDN Reference Tables, https://github.com/dotse/IDN-ref-tables

[MSR] Integration Panel, "Maximal Starting Repertoire - MSR-5: Overview and Rationale", 24-June 2021, https://www.icann.org/en/system/files/files/msr-5-overview-24jun21-en.pdf

Note: MSR-5 is based on Unicode 11.0, the latest version for which IDNA 2008 tables are available at the time of its pubication. Due to the fact that most recent additions to Unicode have been for rarely used code points, the probability that future versions will impact these reference LGRs is very low.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010, https://tools.ietf.org/html/rfc5890.

[RFC6912] Sullivan, A., Thaler, D., Klensin, J., and O. Kolkman, "Principles for Unicode Code Point Inclusion in Labels in the DNS", RFC 6912, April 2013, https://tools.ietf.org/html/rfc6912.

[RFC7940] Davies, K and Asmus Freytag: "Representing Label Generation Rulesets using XML", August 2016 https://tools.ietf.org/html/rfc7940.

[RFC8228] Asmus Freytag: ""Guidance on Designing Label Generation Rulesets (LGRs) Supporting Variant Labels", RFC 8228, August 2017, https://www.rfc-editor.org/info/rfc8228.

[RZ-LGR-Overview] Integration Panel, "Root Zone Label Generation Rules - LGR-4: Overview and Summary", 05 November 2020 (PDF), https://www.icann.org/sites/default/files/lgr/lgr-4-overview-05nov20-en.pdf

[Second-Level Reference] ICANN, Second-Level Reference Label Generation Rules, https://www.icann.org/resources/pages/second-level-lgr-2015-06-21-en