

Integration Panel: Root Zone Label Generation Rules — LGR-1 Overview and Summary

REVISION – December 1, 2015

Table of Contents

| | | |
|----------|---|-----------|
| 1 | Overview | 2 |
| 1.1 | <i>Root Zone Label Generation Rules (LGR-1) Files</i> | 2 |
| 2 | Process of Integration | 4 |
| 2.1 | <i>Overview</i> | 4 |
| 2.2 | <i>Proposals Submitted</i> | 5 |
| 2.3 | <i>Review of Proposals</i> | 6 |
| 2.3.1 | Armenian LGR Proposal | 6 |
| 2.3.2 | Arabic LGR Proposal | 6 |
| 2.3.3 | Contents | 7 |
| 3 | Integration | 7 |
| 3.1 | <i>Merged LGR</i> | 7 |
| 3.1.1 | Repertoire | 7 |
| 3.1.2 | Variants | 7 |
| 3.1.3 | Whole Label Evaluations (WLE) Rules | 8 |
| 3.2 | <i>Arabic Element LGR</i> | 8 |
| 3.2.1 | Repertoire for Arabic | 8 |
| 3.2.2 | Variants for Arabic | 8 |
| 3.2.3 | Whole Label Evaluations Rules for Arabic | 8 |
| 3.2.4 | Default Whole Label Evaluation Rules | 9 |
| 4 | Notes on the LGR | 9 |
| 4.1 | <i>Rules</i> | 9 |
| 4.2 | <i>Scripts</i> | 9 |
| 4.3 | <i>Comprehensiveness and Staging</i> | 9 |
| 5 | Using the LGR | 10 |
| 6 | Limitations of the LGR | 10 |
| 7 | Summary of Changes | 11 |
| 7.1 | <i>Code points by script</i> | 11 |
| 8 | Contributors | 12 |
| 9 | References | 13 |

1 Overview

This document describes the Label Generation Rules for the Root Zone (LGR) developed according to the “[Procedure to Develop and Maintain the Label Generation Rules for the Root Zone in Respect of IDNA Labels](#)” [Procedure]. The Procedure defines a two-stage process, in which community-based Generation Panels (GP) propose LGRs specific to a given script, which are then reviewed and integrated by the Integration Panel (IP). The result of the first round of this development work is the first version of the LGR (LGR-1).

The reader of this document is assumed to be familiar with the [Procedure]¹, particularly the parts that describe the role of the IP and the tasks and expectations on the GPs.

The full content of LGR-1 is specified in a set of files as described in the next section.

1.1 Root Zone Label Generation Rules (LGR-1) Files

LGR-1 is provided as a collection of files that are self-contained and supersede the files from previous versions. The current document (<https://www.icann.org/sites/default/files/lgr/lgr-1-overview-01dec15-en.pdf>) provides background on the content and development of this version of the LGR. It also provides additional guidance to potential users of the LGR.

The normative definition of LGR-1 is provided as a set of XML files, consisting of one merged file and one XML file per script², as shown in Table 1.

Table 1. Merged and Element LGR files (XML)

| Script | File URL |
|---------------|---|
| Merged | https://www.icann.org/sites/default/files/lgr/lgr-1-common-01dec15-en.xml |
| Arabic Script | https://www.icann.org/sites/default/files/lgr/lgr-1-arabic-script-01dec15-en.xml |

The Label Generation rules are expressed using a standard format defined in "Representing Label Generation Rulesets in XML" [XML-LGR]. The merged LGR consists of a list of code points or sequences defining the repertoire as well as a set of mappings providing the variant relations between these repertoire items. In addition, the file contains a merged set of Whole Label Evaluation (WLE) rules for the root zone. Each code point in the file is annotated with the Unicode version in which it was first assigned, and the scripts in which it is used.

Each of the script-specific files contains all the Label Generation Rules applicable to labels from that script, and only those rules. Each file contains a 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.

¹ References to documents cited are provided at the end.

² For LGR-1, with a single element LGR, the two files are nearly identical.

Table 2. Merged and Element LGR files (HTML)

| Script | File URL |
|--------|---|
| Merged | https://www.icann.org/sites/default/files/lgr/lgr-1-common-01dec15-en.html |
| Arabic | https://www.icann.org/sites/default/files/lgr/lgr-1-arabic-script-01dec15-en.html |

For each XML file, a mechanically generated and non-normative HTML presentation, as shown in Table 2, is provided for ease of review. This presentation is augmented by summary data as well as data extracted from the Unicode Character Database [UCD], such as the character name.

Table 3. Other Files

| Contents | File URL |
|----------------------------|---|
| Repertoire Tables, non-CJK | https://www.icann.org/sites/default/files/lgr/lgr-1-non-cjk-01dec15-en.pdf |

A non-normative PDF file shows the repertoire for the majority of scripts in the MSR presented in the form of marked-up tables in a format similar to that used for character code charts in the Unicode Standard:

| | | Arabic | | | | | | | | | | | | | | | |
|---|--|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | | 0600 | 0601 | 0602 | 0603 | 0604 | 0605 | 0606 | 0607 | 0608 | 0609 | 060A | 060B | 060C | 060D | 060E | 060F |
| 0 | | | | | | | | | | | | | | | | | |
| 1 | | | | | | | | | | | | | | | | | |
| 2 | | | | | | | | | | | | | | | | | |

This document shows the code points included in the repertoire for LGR-1 presented in the form of marked up tables in a format similar to that used for character code charts in the Unicode Standard. The background color indicates the status of the code point:

- Green: code points that are part of the LGR.
- White: code points that are not PVALID in IDNA 2008 [RFC5892][IDNAREG].
- Pink: code points that are **excluded** from the Root Zone in a generic fashion (digits, hyphen), or by being excluded from the Maximal Starting Repertoire [MSR-2].
- Lavender: code points not included in the LGR as result of decisions by the Generation Panels during the development of the LGR.

Unicode blocks that contain no repertoire of the LGR are suppressed.

2 Process of Integration

2.1 Overview

The process for developing the Root Zone LGR consists of two stages, whereby a series of community-based Generation Panels creates and submits for public review, a set of Proposed LGRs for their respective scripts. A separate expert panel, the Integration Panel, has the task to select from the submitted LGRs those ready for integration and to assemble them into a version of the Root Zone LGRs.

The [Procedure] assumes that each Generation Panel is best situated to make the selection of code points and variants specific to its script and to propose a disposition for them in the proposed LGR. In general, it is expected that Generation Panels will propose to include only a subset of code points that are in scope for their respective scripts. See also [Guidelines].

The Integration Panel is tasked to evaluate the submitted in light of the Principles laid out in the [Procedure].

The review of LGR proposals undertaken by the Integration panel combines mechanical review steps with qualitative review in light of the principles as described in Section B.4 in [Procedure]. Mechanical review steps include verifying that the proposed LGR

- is within the MSR
- is within the scope (script)
- is symmetric and transitive (with respect to variants)
- contains all default WLE rules and actions
- contains the required files
- meets the syntax requirements

The qualitative review includes evaluation of the proposed LGR against these principles set out in Section A.3.6 in [Procedure] and [IABCP].

Least Astonishment Principle: A Code Point in the Zone Repertoire should not present recognition difficulties to the zone's intended user population and should not lend itself to malicious use.

Contextual Safety Principle: A code point in the Zone Repertoire or any of its Variants that present unacceptable risks of being used in malicious ways should not be permitted.

Simplicity Principle: Overly complex rules are to be avoided, in favor of rules easily understood by users with only some background. In particular, in the root, rules should not require deep familiarity with a particular script or language.

Predictability Principle: People with reasonable knowledge of the topic should by and large reach the same conclusions about which code points should be included.

Stability Principle: Once a code point is permitted, it is almost impossible to stop permitting it: the act of permitting a code point cannot be undone. This is particularly true once a label containing this code point has been registered.

The following principles are normally satisfied implicitly, whether by the way the overall process is organized (by inclusion) or by the way the [MSR-2] defined the boundaries for LGRs. For the inclusion principle, in particular, the IP review checks whether all included code points are justified individually or by being part of a fixed set and documented as such.

Inclusion Principle: The zone repertoire is built up by specific inclusion; the default status for any code point is that it is excluded.

Letter Principle: Only Assigned Code Points normally used to write words should be permitted. Assigned Code Points normally used for both words and other purposes should not be permitted.

Longevity Principle: A Code Point in the Zone Repertoire should have stable properties across multiple versions of Unicode.³

The final principle is an overarching one that applies not only to code points, but also variants and other features of the LGR, and finally to the entire review and integration process. If there are doubts, it is best to withhold approval, rejecting or deferring a proposal until the doubt can be removed. From the Conservatism Principle also follows the prescription in [Procedure] to minimize allocatable variants and to maximize (within reason) the blocked variants.

Conservatism Principle: Any doubt should be resolved in favor of exclusion of a code point rather than inclusion.

The first iteration of this work has been carried out and a Draft Integrated LGR is submitted for public review.

2.2 Proposals Submitted

An integrated LGR starts from proposals for script-based LGRs. At the outset of the work on this LGR the following proposals had been submitted by the respective Generation Panels:

Table 5. Script-Based LGR Proposals for the Root Zone

| Script | Status | File URL |
|-------------------|-----------------|--|
| Arabic | <i>accepted</i> | arabic-lgr-proposal-18nov15-en.pdf |
| LGR Specification | | Proposed-LGR-ArabicScript-18nov15-en.xml |
| Test Labels | | arabic-labels-18nov15-en.txt |
| Armenian | <i>deferred</i> | armenian-lgr-proposal-05nov15-en.pdf |
| LGR Specification | | proposed-armenian-lgr-05nov15-en.xml |
| Test Labels | | armenian-test-labels_05nov15-en.txt |

³ Generally that implies that code points from more the recent versions of Unicode may require more stringent justification for inclusion.

The Integration Panel reviewed these proposals and determined whether they could be integrated into the current version of the LGR.

2.3 Review of Proposals

The Integration Panel was unanimous in accepting the LGR for Arabic for integration into LGR-1. The Integration Panel was divided on accepting the Armenian LGR, because its interaction with other scripts that cannot be evaluated at this time. The proposal is not rejected, but deferred for review in the context of a future LGR.

See the following sections for details.

2.3.1 Armenian LGR Proposal

The Integration Panel worked with the Armenian Generation Panel (ArmnGP) during the development of [Proposal-Armenian]. The Armenian script, being used by a single language with a relatively cohesive community does not pose some of the challenges of more complex or more widely used scripts. However, in developing its proposal the ArmnGP discovered that there exists an interaction with related scripts, based on the existence of cross-script homoglyphs.

Mechanical review verified that the proposal meets all formal requirements.

The proposal for an Armenian Script LGR was thus successfully submitted. Nevertheless, the Integration Panel did not unanimously approve the proposal for integration into the LGR at this time. Unlike the scenarios envisaged in [Procedure] this is not because of a fault in the proposal and does not imply that the proposal should be altered in any way.

The IP has decided, after review of the details of that LGR, that the script should be treated as being related to other scripts in the sense of Section 3.2 of MSR-2. Consequently, the IP chose to defer the script until its interactions with the related scripts are well-enough understood to cause no risk of future incompatibilities.

2.3.2 Arabic LGR Proposal

The Integration Panel worked with the Arabic Generation Panel [TF-AIDN] during the development of [Proposal-Arabic] to ensure that it would meet the Integration Panel's understanding of the [IABCP] principles and other prescriptions found in [Procedure]. In particular, the IP notes the efforts by TF-AIDN to document carefully all code points proposed for inclusion, the steps taken to simplify the specification of variants by removing context rules, and other steps to reduce the number of allocatable variants where that was practical in light of the Simplicity and Predictability principles.

A separate mechanical review of the proposal verified the items in 4.1 and confirmed, by evaluating the supplied test labels that the result of applying the LGR meet the understanding that went into its design.

The Arabic script is a separable script, according to section 3.3 of [MSR-2], and can be integrated in isolation, without the risk of introducing future incompatibilities.

The Integration Panel therefore unanimously approves of [Proposal-Arabic] and selects it for integration into LGR-1.

2.3.3 Contents

As result of the review of proposals submitted, the contents of LGR-1 are defined by a single script-specific LGR, plus the default WLE rules and actions defined by the Integration Panel (IP) as part of the [MSR-2].

3 Integration

After reviewing and accepting each proposed LGR, the Integration panel prepares an XML file containing an equivalent LGR as measured in terms of valid labels and variants produced, but with changes to the metadata and comments for consistency with the other elements of and integration process for the Root Zone LGRs. These constitute the Element LGRs. From each an annotated HTML file is created mechanically for a more human-readable presentation of the data.

From the Element LGRs a merged XML file is created mechanically containing the union of the repertoire and variant mappings and annotating each item in the repertoire and rules with the element LGR that it originates from. Because the actual type of all variant mappings is script-specific and therefore cannot be represented in a merged file, all variant mappings are set to “blocked”. All WLE rules and actions are coalesced. In principle, the WLE rules and actions are not script specific, but in practice they are usually triggered by ranges of code points or variant types specific to an element LGR. The IP manually reviews the result to make sure that rules from different LGRs do not conflict. If necessary, they are restated. Finally, an annotated, human-readable presentation of the merged file is created.

The following sections discuss the contents of particular files making up the Root Zone LGR. These files are listed in Tables 1 and 2 above. For more details and background on the organization of the LGR across files see [Packaging].

3.1 Merged LGR

3.1.1 Repertoire

The repertoire of the merged Root Zone LGR is the cumulative repertoire of all the Element LGRs that have been integrated into this version. Those repertoires, in turn were developed based on [MSR-2], which is a subset of Unicode 6.3 [Unicode 6.3], that excludes code points used for historical or special purposes only, or those used in languages that did not meet the criteria for stable and modern usage as outlined in [MSR-2]

As appropriate for the Root Zone LGR, the repertoire includes neither digits nor the HYPHEN-MINUS.

3.1.2 Variants

The variant mappings in the merged LGR are the union of the variant mappings from all the Element LGRs that have been integrated into this version of the Root Zone LGR. Because the disposition of variant labels, for example as "allocatable", is specific to each script, they cannot be expressed in the

script-neutral context of this integrated LGR. Instead, all variant mappings are given the type "blocked" in the merged LGR.

3.1.3 Whole Label Evaluations (WLE) Rules

The integrated LGR includes the cumulative set of Whole Label Evaluation rules for all Element LGRs that have been integrated into this version. The purpose of WLE rules for the Root Zone LGR is to allow automatic exclusion of labels that present particular challenges in display and processing, such as a label leading off with a combining mark, because that mark would tend to combine visually with the code point in front of it. Based on [Procedure] the Root Zone LGR has a single set of WLE rules that is common to all scripts. In practice, most rules are written to be specific to only the code points encountered in labels of a given script, so that the rules do not interact with each other. Each Element LGR only contains rules that are specified to it (as well as any default rules) while the IP reviewed and made sure that the combined rules in the merged LGR do not give rise to conflicts.

3.2 Arabic Element LGR

3.2.1 Repertoire for Arabic

The repertoire is for Arabic based on Section 3.2 in [Proposal-Arabic] by the Task Force for Arabic IDNs [TF-AIDN] and only includes code points used by languages that are actively written in the Arabic script. It excludes code points for which TF-AIDN was unable to find sufficient evidence of use (see Appendix F in [Proposal-Arabic]).

The Arabic Element LGR does not include combining marks or code point sequences. All combining marks have been excluded for these reasons:

- First, they can significantly overproduce and would require additional rules to contain them effectively, complicating the design.
- Second, even where they are required for some languages, they are optional for others.
- Third, this also circumvents the issue raised by [IAB].

As part of the Root Zone, the element LGR includes neither digits nor the HYPHEN-MINUS.

For further details, see Section 3.2 "Code point repertoire included", in [Proposal-Arabic].

3.2.2 Variants for Arabic

The Arabic Element LGR includes "blocked" and "allocatable" variants, assigned according to Section 4 "Final recommendation of variants for Top Level Domains (TLDs)" in [Proposal-Arabic]. These recommendations balance the desire to minimize the number of possible allocatable variants with the need to keep the definition of variants simple.

3.2.3 Whole Label Evaluations Rules for Arabic

The Arabic Element LGR includes Whole Label Evaluation rules specific to the Arabic script. See Section 5 "Whole Label Evaluation (WLE) rules", in [Proposal-Arabic]. As specified, these rules serve to prevent the mixing of two variants of the same code point within the same label. This has the effect of reducing overproduction of variant labels. See also the comments given for each rule or action.

3.2.4 Default Whole Label Evaluation Rules

The Arabic Element LGR includes the set of required default WLE rules and actions applicable to the Root Zone and defined in [MSR-2].

4 Notes on the LGR

4.1 Rules

“Label Generation Rules (LGR)” is the term used to describe the sets of code points, and the constraints on them, that are needed to generate IDNs in a particular script (e.g. Latin, Arabic, Japanese).

Most of the information takes the form of selections from a repertoire of code points defined in the Unicode Standard. The “R” in LGR stands for “Rules” rather than “Repertoires”, because labels must be constructed out of permitted code points in context, including allowing sequences of code points as repertoire items. The validity of labels is determined by mechanically evaluating the LGR, and in particular the Whole Label Evaluation (WLE) rules, which use the wider context of a label. In addition, variant rules define what variant labels might exist and whether they are or are not available for allocation.

4.2 Scripts

In defining labels fit to be used globally in the DNS root zone, any code point is defined as belonging to a script, with some code points used with multiple scripts, as defined by the Script_Extensions property in the Unicode Character Database [UCD]. For the root zone, all code points used in a given label must normally belong to a single script; although any script supported in the LGR may be used to create a root label, and those labels can in principle be used anywhere in the internet, there cannot be a mixture of scripts represented within a single label. Notably, for example, LGRs for any script other than Latin cannot introduce US-ASCII code points into their labels.

The definition of script for used in the LGR process is that chosen by [ISO 15924]; this definition recognizes that Japanese and Korean are written with a mixture of scripts, respectively a mixture of Han ideographs with Kana or Hangul, and provides separate identifiers for such composite scripts.

Many scripts, such as Arabic, Cyrillic, Devanagari and Latin each support a variety of languages. As long as the code points are members of the same script, as defined by [ISO15924], code points used for different languages can be mixed in a label; subject only to constraint on mixing that might be present in the WLE rules of the respective LGR.

4.3 Comprehensiveness and Staging

Ideally, the LGR would be comprehensive, that is, include all scripts eligible for the root zone from its first version. With respect to the *Stability Principle* and the *Least Astonishment Principle* [IABCP] a fully comprehensive LGR would guarantee that all issues relating to the possible interaction among all scripts can be fully investigated in the development of the LGR. From a practical perspective doing so would be prohibitive because of the additional time needed to investigate certain scripts, and perhaps unnecessary for two main reasons. First, not all scripts are related closely enough so that they affect

each other from the perspective of LGR development. Second, it is not realistic to expect that Generation Panels will be formed and complete their work for all eligible scripts within the same time frame. Consequently, the [Procedure] anticipated that LGR will be rolled out in stages.

The goal for all future versions of the LGR must be to retain full backward compatibility, so that they preserve the output of any label registration against the old LGR, when applied to an updated LGR. Consequently, the IP anticipates that succeeding versions of the LGR will be strict supersets of its predecessors. It is expected that registrations that predate the initial release of an LGR covering the respective script will be allowed to remain, even if in conflict, but without becoming a binding precedent for the LGR itself.

5 Using the LGR

The merged LGR and the Element LGRs serve different purposes. When processing an applied-for label, the context of which script it is to be applied for is specified. That selects the element LGR to use in processing the application. Each script-specific element LGRs presents the complete data and specification to determine the validity of the label as well as generate full set of allocatable variants for the label, when applied for under that script.

By containing the cumulative repertoire, WLE rules and variant mappings (with type set to “blocked”) the integrated LGR thus presents the complete data and specification needed for conflict checking with any existing label, independent of script. Because the validity of a label can in principle be influenced by the definition of variants, the merged LGR cannot be used for final validity checking of a label; in practice, it might serve as a reasonably rough approximation with a tendency to overproduce valid labels.

As much as possible, the variant mappings and types have been drawn up to limit the number of allocatable variants generated. Where applicable, WLE rules reduce the number of valid labels, and in some cases they reduce the number of allocatable variants as well. Both mechanisms rely on dividing the variants by linguistic context and on mechanically preventing the mixing of variants from different contexts in the same label.

In accordance with the [Procedure] the LGR is designed to mechanically eliminate (or take off the plate) as much as possible any labels and variant labels that pose an undue risk to the usability and security of the DNS.

6 Limitations of the LGR

There are limitations to what can be done with mechanical application of rules, and in some cases, it is not possible to reduce the number of allocatable labels that is practicable and safe without creating undue restrictions on otherwise valid labels. In this context it is a useful reminder that having a label that is “allocatable” neither means that it will necessarily be delegated, nor that it necessarily should be delegated. In fact, investigations of actual registrations on the second level reveal that applicants have tended to apply for only a small number of variant labels.

The LGR can be thought of as creating a maximal set of valid labels and allocatable variants, but other steps in the registration process are expected to include suitable mechanisms to shortlist the set of labels for delegation. It is the view of the Integration Panel that such shortlisting is absolutely necessary, because increasing numerocity of delegated variant is concomitant with an increased risk to the DNS.

7 Summary of Changes

LGR-1 is the initial LGR.

1. LGR-1 added 128 code points for 1 script, 17 WE rules and 21 actions.

7.1 Code points by script

The following table shows how many code points, by script, are available for root zone LGR development by being included in [MSR-2] and how many are selected for each version of the LGR.

| Script tag | Script Name | MSR-2 | LGR-1 |
|--------------|----------------|--------------|-------|
| Arab | Arabic | 239 | 128 |
| Arm | Armenian | 38 | |
| Beng | Bengali | 64 | |
| Cyrl | Cyrillic | 93 | |
| Deva | Devanagari | 91 | |
| Ethi | Ethiopic | 364 | |
| Geor | Georgian | 37 | |
| Gre | Greek | 36 | |
| Gujr | Gujarati | 66 | |
| Guru | Gurumukhi | 61 | |
| Hang | Hangul | 11172 | |
| Hani | Han Ideographs | 19852 | |
| Hebr | Hebrew | 46 | |
| Hira | Hiragana | 89 | |
| Kana | Katakana | 92 | |
| Khmr | Khmer | 78 | |
| Knda | Kannada | 68 | |
| Laoo | Lao | 53 | |
| Latn | Latin | 305 | |
| Mlym | Malayalam | 73 | |
| Mymr | Myanmar | 90 | |
| Orya | Oriya | 66 | |
| Sinh | Sinhala | 79 | |
| Taml | Tamil | 49 | |
| Tel | Telugu | 67 | |
| Thaa | Thaana | 50 | |
| Thai | Thai | 71 | |
| Tibt | Tibetan | 80 | |
| Zinh | INHERITED | 21 | |
| Total | | 33490 | |

8 Contributors

LGR-1 was developed by the Integration Panel, based on proposals submitted by the respective Generation Panel, with input from community members, as well as support by ICANN staff members.

1. Integration Panel Members

Marc Blanchet
Asmus Freytag
Nicholas Ostler
Michel Suignard
Wil Tan

2. Advisors

Lu Qin

3. Community Members

The integration panel gratefully acknowledges the information provided by the following members of the community:

Members of TF-AIDN
Members of the Armenian Generation Panel
Chris Dillon
Olivier Crepin-Leblond
Yuriy Kargapolov
Narine Khachatryan
Meikal Mumin
Dusan Stojicevic

4. ICANN Staff

Sarmad Hussain
Alireza Saleh
Anand Mishra
Jia-Juh Kimoto
Kim Davies

9 References

- [Guidelines] Internet Corporation for Assigned Names and Numbers, "Guidelines for Developing Script-Specific Label Generation Rules for Integration into the Root Zone LGR". (Los Angeles, California: ICANN, December 2014)
<https://community.icann.org/download/attachments/43989034/Guidelines-for-LGR-2014-12-02.pdf>.
- [IAB] Internet Architecture Board (IAB), "IAB Statement on Identifiers and Unicode 7.0.0"
<https://www.iab.org/documents/correspondence-reports-documents/2015-2/iab-statement-on-identifiers-and-unicode-7-0-0/>
- [IABCP] Sullivan, A., *et al.*, "Principles for Unicode Code Point Inclusion in Labels in the DNS". Internet Architecture Board (IAB) = RFC 6912
<http://tools.ietf.org/html/rfc6912>.
- [IDNAREG] IANA Registry: "IDNA Parameters". For Unicode 6.3 available at:
<http://www.iana.org/assignments/idna-tables-6.3.0/idna-tables-6.3.0.xml>. Visited 2013-11-20.
- [ISO15924] *Codes for the representation of names of scripts*, ISO 15924:2004. Available from
<http://www.unicode.org/iso15924/>. Visited 2012-09-21.
- [MSR-2] Integration Panel, "Maximal Starting Repertoire — MSR-2 Overview and Rationale", 14 April 2015 <https://www.icann.org/en/system/files/files/msr-2-overview-14apr15-en.pdf>
- [Packaging] Integration Panel: "Packaging the MSR and LGR", 24 April 2015,
<https://community.icann.org/download/attachments/43989034/Packaging-MSR-LGR.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>
- [Proposal-Arabic] TF-AIDN, "Proposal for Arabic Script Root Zone LGR", Version 3.4, November 18, 2015 <https://www.icann.org/en/system/files/files/arabic-lgr-proposal-18nov15-en.pdf>
- [Proposal-Armenian] Armenian Generation Panel, "Proposal for Armenian Script Root Zone LGR", November 05, 2015 <https://www.icann.org/en/system/files/files/armenian-lgr-proposal-05nov15-en.pdf>

- [RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.
- [RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.
- [RFC5892] Faltstrom, P., Ed., "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)", RFC 5892, August 2010.
- [RFC5893] Alvestrand, H., Ed., and C. Karp, "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)", RFC 5893, August 2010.
- [RFC6912] Sullivan, A., *et al.*, "Principles for Unicode Code Point Inclusion in Labels in the DNS", RFC 6912, April 2013. = IABCP
- [TF-AIDN] Blog, "Task Force for Arabic Script IDNs" <https://www.icann.org/news/blog/what-is-the-task-force-on-arabic-script-idns-tf-aidn-up-to>
- [Unicode63] The Unicode Consortium. The Unicode Standard, Version 6.3.0, defined by: "The Unicode Standard, Version 6.3.0", (Mountain View, CA: The Unicode Consortium, 2013. ISBN 978-1-936213-08-5). <http://www.unicode.org/versions/Unicode6.3.0/>.
- [UCD] UAX #44: *Unicode Character Database*. An integral part of The Unicode Standard. Most recent version available from <http://www.unicode.org/reports/tr44/>. Version 6.3 available as <http://www.unicode.org/reports/tr44/tr44-12.html>.
- [XML-LGR] Davies, K. and A. Freytag, "Representing Label Generation Rulesets using XML", <http://tools.ietf.org/id/draft-ietf-lager-specification-04.txt>