Procedure to Develop and Maintain the Label Generation Rules for the Root Zone in Respect of IDNA Labels

First public draft (Version 2012-09-24c)

Table of Contents

A. Introduction, Motivations, and Background

A.1. Purpose of this section

This document defines procedures for creating and maintaining part of the label generation rules for the root zone. The resulting label generation rules will provide a consistent and predictable set of permissible code points for IDN TLDs and provide a way to determine whether there are variant labels (and if so, what they are). For the purposes of this document, the label generation rules contain four parts: the rules governing the permissibility of Unicode code points (the repertoire), any exchangeable code point variants that follow from those (the variant rules), the status of any resulting label, and a set of optional whole-label evaluation rules that determine whether the output of the previous three portions is still an acceptable label in the root zone. This document defines the *procedures*, and the label generation rules themselves. This section presents some important background and some motivations for the procedure that is proposed later in the document.

A.2. Conventions and background

There is a bibliographic list in section E. We put references in square brackets, using what we hope is a meaningful short name for the item. When the same reference is used in running text, we use the short name without the square brackets. We refer to publications in the Request for Comments series by their RFC number, even if they are part of some other series (BCP, STD, and so on). Unicode documents are referred to following the Unicode convention, where the number sign is included in running text but not when used in a reference (so, "UTR#36", but "[UTR36]").

The following are prerequisites for understanding this document:

- "A Study of Issues Related to the Management of IDN Variant TLDs (Integrated Issues Report)" [IIR];

- IDNA2008 [RFC5890] [RFC5891] [RFC5892] [RFC5893] [RFC5894] [RFC5895];
- Unicode [Unicode61];
- "Principles for Unicode Code Point Inclusion in Labels in the DNS." [IABCP].

In addition, the terms defined in Appendix 2 of IIR are incorporated here by reference, and not reproduced. *Some of the terms defined in that Appendix 2 are used in a special or peculiar way*, and the text below is unlikely to be understood completely without having that terminology to hand. (We have not followed the capitalization convention of IIR, because some readers found it confusing, but the terms are otherwise the same.)

The draft sometimes includes discussion of things like "all of Unicode". That phrase is really an abbreviation for "all of the parts of Unicode that are somehow permitted under IDNA2008." Some parts of Unicode are already not permitted by IDNA2008. Anything that is not permitted by the protocol at all is automatically removed from consideration.

A.3. Label generation rules

Every zone on the Internet has, either implicitly or explicitly, a set of rules governing the labels allowed in that zone. Sometimes, these are implicit and trivial, such as, "I only permit labels that have something to do with my company," or, "We name all our hosts after moons of Jupiter." Sometimes, they are more involved: many TLDs have exclusion lists of labels that are not permitted. In the root zone today, two-character ASCII labels are either withheld or else allocated to qualified entities (the country of the country code in question). We call all such rules the label generation rules. This document provides a mechanism in order to generate rules necessary for long-term operation of both IDNs and IDN variants in the root zone.

A.3.1. Some starting premises

We start with the premise that it is beneficial for the Internet community to permit some labels conforming to IDNA2008 in the root zone. We acknowledge that not all agree with this premise, but observe that the root zone already contains such labels, and conclude therefore that label generation rules are needed to govern such additions to the root zone. Moreover, the addition of U-labels and A-labels to the root is in keeping with DNS names being useful mnemonics: it is very hard to remember a name written using characters one does not normally use. The basic, positive good that ought to come from a new set of label generation rules is a basic repertoire of (assigned) Unicode code points that can be helpful in building usefully-mnemonic labels in the root zone. This does not mean that every word or string – or even most such strings – from a language will be eligible under the new label generation rules; only that useful mnemonics can be represented. There is no intent

for the procedure and the rules it produces to support fully (or fail to support fully) any particular living language community.

A.3.2. Variants

With the addition of U-labels and A-labels to the root zone, the question of IDN variants inevitably follows. As noted in IIR (footnote 26, p 41), IDN tables (as described in RFC3743 and generalized in RFC4290) are implementations of label generation rules; they include two separable but linked components. These components are the code point repertoire for the zone (or zone repertoire), and the code point variant rules. Those two components are part of the label generation rules.

IIR argues that, because the root zone is necessarily shared by everyone on the Internet, it is a special case, and needs a set of rules that ensures minimal conflict, minimal risk to all users (as opposed to users of one or another language or script), and minimal potential for incompatible change over time. These conclusions are in keeping with ICANN's responsibility for the security and stability of the root zone.

A.3.3. Characteristics of the Process Goals

The process can be further characterized by the following. Because of the interest in IDNA labels for the root, the intent of the process is to move as expeditiously as possible within the bounds of the safety and security for the root zone and in keeping with expert assessment. Beyond the interest of security the following four main characteristics apply:

Utility

Even though it is not a goal to support anything that can be written as a word in a particular language as a label, the mnemonics allowed by the label generation rules should have a certain utility to them. This would not be satisfied by very arbitrary restrictions of the repertoire or the code point variant rules.

Coverage

The coverage of the repertoire should be comprehensive, so as to not exclude a certain script community;

Non-arbitrariness

Provisions in the LGR should not be arbitrary in the sense that security concerns are evaluated more tightly for one community over another.

Absence of bias

The procedures and their results should not be biased in the sense that they only care about communities based on some criteria such as overall size, representation by a national government or similar factors.

These are characteristics of the *goals* of the procedures, and not necessarily of the procedures themselves: it may be that no procedure meets every one of these goals separately, and that any given language community may be affected negatively in order to achieve a satisfactory result overall.

A.3.4. Precedents

Even if there are existing precedents based on existing TLDs, the procedure in this document establishes a new playing field. While existing labels will almost certainly have to be grandfathered even if they are in conflict with the label generation rules established by this procedure, that precedent, and that conflict is not a reason to invalidate any aspect of the new rules or this procedure.

A.3.5. Principles to constrain the label generation rules

IIR's argument is consistent with views expressed in an Internet Architecture Board Internet Draft [IABCP]. IABCP lays out several principles that are useful guides for (or perhaps constraints on) developing the label generation rules for any zone, including the root zone. The principles relevant to the root zone are paraphrased below, but the reader should consult IABCP for a full discussion. Note that, for the purposes of this document, a code point in the zone repertoire is necessarily an assigned code point.

The principles are not rules; they inform, but cannot substitute for the judgment of those making decisions in the label generation rules development process. They are intended to be weighed in the context of the project goal, premises and characteristics outlined above.

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

For characters recently added to Unicode, there is a risk that as use of that character grows, refinements may need to be made in certain aspect of its recommended use. This principle intends to minimize that risk.

Usability 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.

Contrary to the generally understood meaning of this term, this principle addresses only the risk to usability of a code point due to abuse or recognition difficulties.

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

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.

An overall familiarity with basic concepts of Unicode, for example, would be acceptable background; rules based on such concepts, applied universally across scripts would not necessarily violate that principle.

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.

This principle addresses the risk introduced by the "sticky" nature of the label generation rules and expresses the risk it introduces. Choices that are least likely to be overturned by later review or future additions are preferable to those that may be overturned.

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.

The letter principle intends to restrict the repertoire to the space needed for mnemonics, without relying on the Unicode letter property which is drawn both too narrowly and too widely. The principle does not state that all letters must be permitted.

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

There is a sense in which the conservatism principle can be considered as an overriding principle. It requires that all decisions to include a code point or rule be based on strong agreement and high confidence.

All of these principles, excepting perhaps the Letter Principle, are easily applied to code point variant rules as well, and therefore are relevant to the project's work. We take these principles to be foundational ones in evaluating the risks of any plan.

There is a certain amount of tension between even similar principles. For example, reviewing a recently added code point would be at odds with the longevity principle but, by being able to evaluate it in the full context, might prevent the addition of a rule that would be in conflict if the code point was added in the future, thus satisfying the stability principle.

IABCP makes plain that it is talking about principles, and not algorithms. If there were an algorithm (or indeed any automatable system) for determining the label generation rules, it is reasonable to suppose the IAB would recommend using it. We must therefore conclude that, whatever the procedure to create and maintain the rules is, it will involve humans and judgment.

A.3.6. Evaluation Parameters

IIR outlined (pp 42-43) three independent parameters that could be used in evaluating the proposed process for label generation rules. They are comprehensiveness, expertise, and qualification. In the discussion arriving at the proposed process outlined below, we have identified a fourth parameter, centralization.

Section D, Evaluation of this Proposal Against the "Parameters" describes these four parameters in detail and uses them to evaluate the proposed process.

B. A development methodology

In this section, we outline a methodology for building and maintaining the root zone label generation rules relevant to IDNs in the root[1].

The methodology recognizes that, while the zone repertoire and code point variant rules are logically separate, in practical terms if there are to be variants they need to be considered at the same time as the repertoire.

We believe that core elements of the development methodology are also applicable to a later, perhaps less intensive maintenance phase, because the basic similarity of the mechanics of both development and maintenance.

---

[1] Much of this document ignores the distinction, but strictly the label generation rules that come from this procedure apply only to IDN labels. Other label generation rules already exist, and we expect they will persist. For instance, all 2-character LDH-labels are currently withheld, or else they are delegated as country code TLDs (based on the country named by that two-letter country code).

B.1. Overview

A repertoire that satisfies the Longevity, Usability, Predictability, and Stability Principles, cannot be built *ad hoc*. In order to conform to the Usability Principle, it will be necessary to bring to bear expertise in the writing systems appropriate to a range of code points. But, given the user population for the root zone (i.e. everyone on the Internet) and the Stability Principle, the final repertoire must have a unity that will not come from considering subsets of Unicode independently. We therefore envision a two-pass process.

B.1.1. How the output of this procedure is to be consumed

The operation of this procedure yields all the label generation rules for the root zone that apply to IDN labels. Any application for an IDN label in the root zone is evaluated using the label generation rules. This step is fully automatic, so the label generation rules that are the product of the procedure in this document must always yield a single answer for the entire candidate label, (See section B.1.2.1 for more on this.)

B.1.2. Output

The output of the process will be a unified set of label generation rules that specifies the following:

1. The complete list of code points permissible in U-labels in the root zone (the zone repertoire);
2. The complete code point variant rules for each code point (if any) for generating variant labels. For any code point in the repertoire, this element of the label generation rules gives a complete list of all the other code points (including series of code points) that are variants for the first. This element is also used to specify variants for a structured series of code points.
3. The disposition of the labels resulting from the application of the rules in (2). For a given code point in the repertoire, this element specifies whether a resulting variant label is blocked, withheld, or allocated.
4. Assignment of one or more "tags" to each code point in the repertoire, and to each disposition of a variant in (3). For the same variant label, different dispositions may exist, as long as they do not share common tag values.
5. A final, optional, whole-label evaluation rule that determines whether the original, applied for candidate label as well as each of its variants is permitted in the root zone.

Different scripts (and sometimes, different languages using the same script) require different treatment, so the output can generate labels and handle variants differently based on script, but also language. The mechanism to accomplish this differential treatment is the tag assigned in item (4); these are described further in

section B.3.1.1. Note, however, the code point variant rules (in 2, above) are always exactly the same for any given code point. This way, code points are always related to one another in a predictable way. For the same reason, the whole-label evaluation rules are always the same for any given code point, and are also the same for every code point in a single script.

All code points sharing a specific tag will implicitly form a subset of the repertoire, not necessarily disjoint of all other such subsets. Labels will be restricted to code points from any one of these subsets only. Disposition of labels will be based on the tag associated with that subset.

Wherever possible, subsets and dispositions based on script are preferable, but some situations call for language-specific rules.

B.1.2.1 Operating the resulting label generation rules

Because the output of the secondary panel is required to be internally consistent, it is necessary that the label generation rules be operable by way of four functions (which could be combined into a single, larger function as a matter of implementation):

1. A function that operates on the LGR and takes a candidate TLD label as input and determines unambiguously all the variants (if any) of that candidate label;
2. A function that operates on the LGR and takes a candidate TLD label as input and determines whether there is a match / conflict between it and any variant of any allocated TLD label, or of any currently requested TLD label;
3. A function that operates on the LGR and takes a candidate label as input and determines unambiguously which of the variants of that label must be blocked or withheld and which may be allocated in the root zone.
4. A function that operates on the LGR – specifically, the final whole-label rule portion – and takes a candidate label as input and determines unambiguously whether any given candidate label (including output of the foregoing steps) is permitted in the root zone.

These four functions provide the necessary conditions for the functioning of the label generation rules.

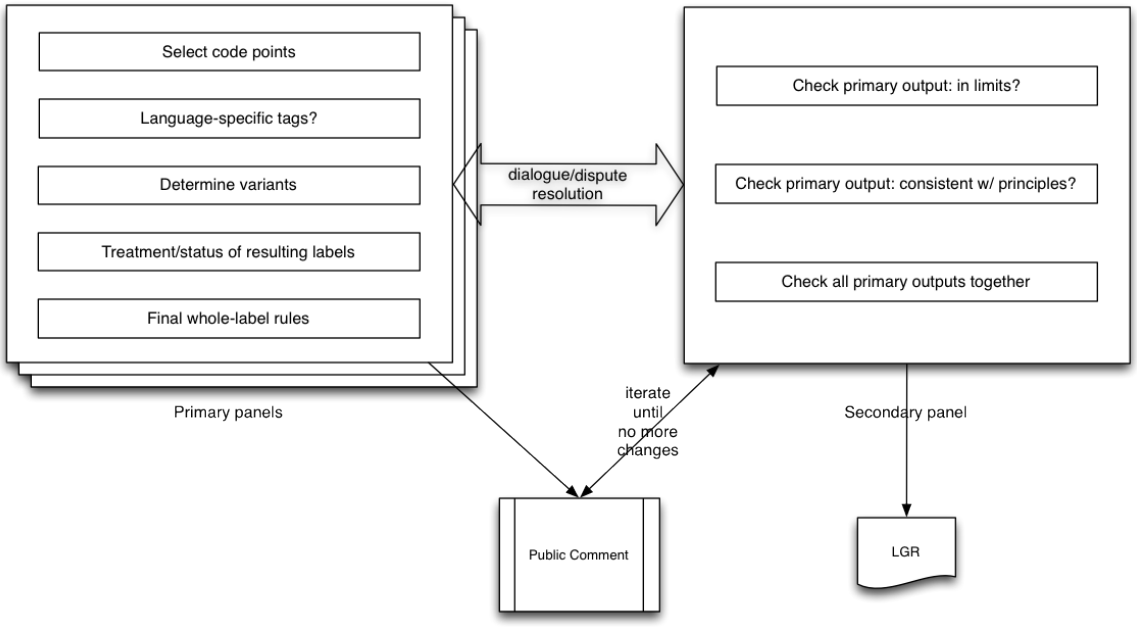
B.1.3. Two-pass process

The first pass involves people interested in a given script, writing system, language, or all of these; they produce a set of label generation rules relevant to that interest. The second pass involves a unified expert panel that creates a common set of label generation rules for the root zone out of the input from the first panels. These two panels are assisted by technical advisors, who observe the activities of all active panels; who provide comment and advice on the technical matters of IDNA, Unicode,

DNS, and linguistics; but who do not otherwise have a formal role in making a decision. In what follows, we call the first of the panels the "primary panel", and the second of them the "secondary panel."

In order to treat matters as generically as possible, the descriptions that follow do not make reference to specific examples. Neither does this section provide specific advice to panels. Some possible scenarios are discussed in detail in the appendices.

The diagram below is a high-level description of the process.



B.2. Establishment and Composition of Panels

B.2.1. Primary Panel

Each primary panel works on a subset of Unicode relevant to one writing system or a set of related writing systems. This work is broadly aligned along the script property of the Unicode code points in question, though it need not be restricted to a single script (see Section B.5.4).

B.2.1.1 Establishment

A primary panel would normally be organized only if there is interest from some linguistic community. In addition, for the root zone, any language or script that is not in active, everyday use by a living community of speakers is barred from consideration, even if some parties ask that it be supported. This is in keeping with the Conservatism Principle, because it discourages more abstruse cases and only supports the real population of zone users (i.e. the population of Internet users).

B.2.1.2 Expertise

All primary panels should have some expertise in the writing systems concerned, but need have neither overall expertise, nor expertise in any other writing system. Panels may be made mostly of volunteers interested in that portion of the potential repertoire. When ICANN determines that a primary panel is needed, it appoints a chair for the primary panel.

B.2.1.3 Diversity

Primary panels need to have some diversity of participation in order to be useful. They must have sufficient numbers of participants so as to ensure that the work of the panel is not all essentially that of a single person. In addition, participation should be diverse in sponsorship: a panel of 10 people all employed by the same organization is insufficiently diverse to qualify. When establishing a primary panel, ICANN makes a public call for participation in the work of the panel. ICANN may also appoint members of the panel. If it is impossible to get sufficient numbers of participants with sufficient diversity, that is evidence that the code points in question and the rules associated with them are too specialized to be included in the root zone, and so the primary panel will not be established.

B.2.1.4 Work plan

In order to avoid later doubt arising from insufficient diversity, at the time a primary panel is set up, it produces a short, rough work plan for the portions of Unicode that the panel expects to deal with. The secondary panel reviews that work plan and the résumés of the primary panel members. The secondary panel may make recommendations for additional participation, or additional or reduced scope of the primary panel's work. The recommendations are advice, not requirements, although they are an indication that the secondary panel may later use its reservations about the participation in the primary panel as one of the reasons for the secondary panel's doubts.

B.2.1.5 Sub-panels

Primary panels, particularly those concerned with scripts used for a wide variety of languages, might prefer to organize their work into sub-panels. Such additional organization of work is beyond the scope of this document, and we consider such a decision to be entirely a management decision on the part of the primary panels (so it is neither hereby forbidden nor encouraged). A primary panel that attempted to organize its work this way might find it desirable to echo the relationship between the primary panels and the secondary panel in the relationship between the sub-panels and the primary panel. Subpanels may include participants from outside the primary panel itself (for instance, writing system experts with particular expertise for some languages).

B.2.2. Secondary Panel

The second pass is carried out by a unified expert panel that reviews all available first-pass output and recommends a resulting set of label generation rules. The secondary panel reviews the output of all available primary panels, taking that output as proposals for parts of the final set of all label generation rules. The secondary panel must produce, each time it completes a round of work, a single, complete, unified, and internally consistent set of rules.

B.2.2.1 Composition

This secondary panel should be made entirely of ICANN-paid consultants or staff (or both) duly qualified against conflicts of interest, in order to minimize any appearance that the process might be captured by interested parties; and so that ICANN may require, as part of its contract with the panelists, disinterested evaluation. The panel requires at least one expert in Unicode issues, at least one expert in IDNA and DNS issues (or one for each), and at least one expert in linguistics and writing systems (who could be the same as the first expert, but need not and often will not be). No member of any primary panel may be a member of the secondary panel.

B.2.3. Advisors

Any panel, at its discretion, may call on advisors. Advisors have particular expertise in issues relevant to the questions at hand, and are there to provide observations given that expertise. Advisors may be paid consultants, ICANN staff, or volunteers. They have no formal role in the decisions of the panels, though their expert opinions may be influential on the panelists. Anyone taking an advisory rule is automatically excluded from participation in any primary or secondary panel. It seems likely that many ICANN staff, which will need to participate in the work of both the primary and secondary panels, will fall into the category of "advisor".

B.3. Procedure

B.3.1. Primary panel proposals

The basic job of the Primary panel is to produce one or more proposed lists of code points to be included in the zone repertoire, and any associated code point variant rules for those code points. Normally, a primary panel will produce one such list, but in some circumstances it might need to provide more than one.

When primary panels are established, they are chartered to cover some section of the Unicode character repertoire, corresponding to the usual way of writing some language or group of languages. Normally, the section of the Unicode character repertoire corresponds to the script property of the code points, with a few inclusions of code points having the Inherited or Common property. Because the

portions of the Unicode character repertoire do not align perfectly with any pre-existing classification of characters, part of the work of the primary panel is to determine which of the relevant code points to include in the root zone repertoire, and which resulting code point variant rules are necessary.

B.3.1.1 Naming the primary panel output: tags

By necessity, the primary panels will consider a subset of Unicode, and because some Unicode characters are shared between scripts, these subsets will, in the general case, not be fully disjoint. At the same time, for linguistic reasons, it may be necessary to give different disposition to variant labels depending on script or language. For this reason, the output of the primary panel is associated with a descriptive identifier, called a tag. (In some cases, a primary panel might be responsible for the creation of output for more than one linguistic context, and each would get a separate tag).

These tags are used at the time of application for a U-label in the root zone, in order to identify the relevant portion of the repertoire to consult. Every code point in the applied-for U-label must be included in the portion of the repertoire named by the tag, or else the application is invalid. Also, the set of variant labels that arise from the applied-for U-label is calculated using the rules identified by the tag, and the resulting status of each such label is also determined by the variant rules the tag names.

In most cases, repertoire subsets consist almost entirely of one script. The name for that subset and associated rules would be based on the name of the script. In more unusual cases, a distinction on some other basis (most commonly language) is necessary.

The requirement that all code points for an applied U-Label fit into the repertoire corresponding to the tag submitted with the applications allows the process to administer restrictions, such as forcing labels to not be of mixed script, unless that is specifically allowed for a given tag.

The following describes the detail of how to create tags based on script or language names.

B.3.1.2 Structure of tags

The “tags” are language tags as described in RFC 5646, with an extension to identify the tag as applying to the root DNS zone. They are composed of the following parts:

1. A language subtag, as specified in the language subtag registry maintained by IANA and created by RFC 5646;
2. A script subtag, using the appropriate ISO 15924 identifier;

3. A singleton subtag, to be assigned by IANA – in this document, it is styled TBD, but it will be a single lower-case ASCII letter when it is assigned;
4. The extension subtag "root", to indicate that this named set of label generation rules is for the root zone.

In the normal case, a primary panel develops label generation rules for one or more scripts. In this case, the language subtag used is "und", and the script subtag is the ISO 15924 identifier appropriate to the script. This is the normal case because DNS labels are never really in any language. In this case, the zone repertoire named by the tag is the repertoire for any label using that script. For any given code point in the named repertoire, the code point variant rules for the code point generate the very same list of variants, and the operation of any given code point variant rule always produces a variant label with the same status, no matter what the linguistic environment of the applicant is.

The use of some labels is sensitive to the intended user population's language, and in those cases it is possible to use a tag starting with a different language subtag than "und". In this case, there may be more than one tag for the same script. The different tags may have different repertoires, and they may result in different status values for the resulting variant labels.

B.3.1.3 Restrictions and Rules for assigning Tags

The use of tags with a specific language subtag (other than "und") comes with three restrictions:

- If a primary panel wishes to use a language specific tag for a script, there must exist more than one language subtag for the same script. If not, there is no reason to use anything other than und, and define the label generation rules for the script.
- If there is both a generic tag (with "und" as the language subtag) and a tag with a specific language subtag for the same script, the repertoire named by the language-specific tag must be a subset of that named by the generic, script-based tag. In other words, the tag that covers the most general use of the script must in fact be the most general.
- If there is a generic subtag and a language-specific subtag for the same script, the language-specific subtag may make a variant status more restrictive, but not less. So, if the generic subtag's code point variant rule results in a label that is allocated, a language-specific subtag's code point variant rule might result in a label that is blocked. By way of contrast, if the generic subtag's code point variant rule results in a label that is blocked, the language-specific subtag cannot make the label allocated.

In addition, there is one overarching restriction about code point variant rules. The code point variant rules for a given code point must generate the same list of

variants no matter what the tag in use. So, if a code point is part of the repertoire named by three different tags, the code point variant rule in all three produces exactly the same list of variant code points. These variants may differ only in respect of their status: one tag may name the rule where all the variants are blocked, and another may name the rule where some of the variants are allocated; but in every case, the list of variants is always the same for a code point.

B.3.1.4 Unicode Script Mixing

This procedure makes it possible that a label will mix code points with different script properties in a single label. The mechanism of using named repertoires with corresponding rule set allows in principle the creation of mixed script repertoires. At the same time, it assures that any script mixing has to be explicitly allowed for, by creating a specific named repertoire that contains only the particular code points whose scripts may be mixed, and only for applications that use the corresponding tag. Thus, the procedure ensures the spirit of the Inclusion principle is applied – any mixing that is to be allowed has to be deliberately included via a named repertoire.

It is anticipated that, based on the Principles, script mixing would be normally be restricted by the secondary panel, rather than allowed to be applied widely. Mixing of characters with the COMMON or INHERITED script property would probably have to be allowed in the appropriate contexts, as well as mixing of the Hiragana and Katakana syllabaries with Han for Japanese. Beyond that, we anticipate that such mixing would mostly be restricted, because of the systemic risks it presents. Detailed rules are outside the scope of this document, which is properly concerned only with defining a suitable procedure within which determinations can be made.

B.3.1.5 Final whole-label evaluation rules

There are some sequences of code points that can be typed, but which are structurally ill-formed. This phenomenon is particularly important in complex writing systems. For such cases, the primary panel may propose a set of final whole-string evaluation rules. When the label generation rules are used, these whole-label evaluation rules are applied after all the other steps, to test all the resulting variant labels as well as the original, applied-for candidate label. If any label does not meet the tests in the whole-label evaluation rules, that label is automatically blocked notwithstanding any other result from the label generation rules.

The whole-label evaluation rules are attached to the code points in a named repertoire subset. Every code point named by a tag is attached to the same whole-label evaluation rules. In principle, this means that it is possible for inconsistent whole-label evaluation rules to be defined; it is up to the secondary panel to ensure such a condition does not arise.

A sensible place to start in building the final whole-label evaluation rules is UAX#29. UAX#29 develops the notion of grapheme clusters, and a useful generic whole-label evaluation rule might be a requirement that labels only ever be made of integral Default Grapheme Clusters. More complicated rules might be required, depending on the script in question; such determinations should be made by the primary panel and confirmed by the secondary panel.

B.3.1.6 Steps

The primary panel's starting point is the repertoire of Unicode characters needed for the writing systems in question, already reduced based on IDNA2008 and the principles in IABCP. Normally, a primary panel starts with the reduced list of code points consisting of all the Assigned Code Points likely to be used. The list is usually made up of code points with the same script property, plus relevant code points with the Inherited or Common properties.

To complete the first task, the primary panel shall also exclude from the repertoire all characters defined as restricted for identifiers, as specified in Table 1 of UTS#39. (See also Section B.5.4.)

The panel's second task will be to start with the alphabet repertoires of the languages used by the panelists, since these will obviously be of interest to them. Typically there is a base set shared by most languages, together with local extensions for specific languages.

The third task will be to look at those code points from the original list that are not indicated in the second task. The panel must exclude any code points used only for archaic or historical purposes (for example in medieval manuscripts). It must further exclude all code points used as special phonetic or other notational characters unless they are in current use in a natural orthography. (For example, there is considerable overlap between the International Phonetic Alphabet and orthographies in Africa and the Americas.)

To complete the third task, the panel shall next consider each of the remaining code points and establish whether other alphabets can be identified authoritatively as containing these code points, making them eligible for inclusion into the repertoire. Any character that cannot be authoritatively established as being used for everyday writing in a living language will be excluded. For this determination the panel may rely on outside expert knowledge; but it is the responsibility of a primary panel to come to a definite conclusion whether it is both safe and useful to select a particular code point as candidate for the repertoire. In case of uncertainty or doubt about the expertise or the authoritativeness of the information available, the panel must exclude the character as candidate.

Having proceeded through these, the panel addresses each included code point in turn, and determines whether there are any code point variant rules for the code point. The variant rules from the primary panel fall into three categories:

1. Code point substitutions, 1:1;
2. Code point substitutions 1:many or many:1;
3. Code point substitutions of one series of code points for another series of code points, but only in cases where the writing system needs it due to features of that writing system. This category explicitly excludes any case where the many:many relationship cannot be treated as completely automatic.

Any rule that depends on context will require very strong evidence that it is in fact required to write any useful mnemonics for some language users; the primary panel shall proceed on the presumption that a code point that requires context rules is likely to violate the Simplicity, Conservatism, and Usability principles. In any case, code points permitted by IDNA2008 under the CONTEXTO and CONTEXTJ rules are automatically excluded.

Part of the specification of the code point variant rules involves specifying the resulting treatment of variant code points, in order to determine whether a code point substitution results in an active, allocated, withheld, or blocked variant. It is possible that the only difference between two tagged portions of the repertoire will be the resulting treatment. See section B.3.1.1 for more discussion.

Finally, the primary panel will define any final whole-label evaluation rules necessary for the named repertoire.

The primary panel, when it has completed its work, sends its recommendations to the secondary panel. At the same time, the primary panel's recommendations are posted for public comment using the prevailing ICANN public comment procedures of the day.

B.3.2. Secondary panel review

The secondary panel reviews the output of every primary panel from which a proposal is available at the time the secondary panel begins review. The secondary panel evaluates each proposal. The secondary panel first confirms that the proposal stays within the maximal repertoire defined as the starting point by the secondary panel, and that it conforms to the other requirements for output set forth in this document.

It then evaluates the proposal for consistency with the Principles and for the risk it presents, in the context of the entire starting repertoire of Unicode code points. Proposals that do not meet the principles or create unacceptable systemic risk are

rejected. A primary panel's proposal may be to permanently exclude a code point, which would factor in this evaluation.

The secondary panel further ensures all proposals together form an internally coherent set of label generation rules, and rejects any proposals that are in conflict. In particular, it makes sure that proposals tagged with language-specific tags are a proper subset of at least one script-tagged proposal. Finally, it reviews whether the proposals collectively meet any other requirements (symmetry etc.) set forth in this document. Proposals that fail this review will be rejected.

Unlike the primary panels, the secondary panel will consider possible interactions with Unicode characters outside the proposed set of rules and, if necessary, reject the primary panel's proposal based on such issues. The secondary panel includes in its evaluation possible sets of rules that are not yet proposed, as might happen when a writing system did not initially attract a primary panel but might be expected to attract one in future. Decisions by the secondary panel are required to be unanimous; any proposal that does not attract unanimous acceptance is automatically not accepted.

B.3.2.1 Conflicts with the primary panel

In case the secondary panel rejects a proposal by a primary, the panels involved must negotiate agreement. The panels must reach consensus on any label generation rule for it to be included. This is true even for the resulting treatment, such that a secondary panel might agree that the code point in question should be included in the zone repertoire, but that the resulting treatment of other code points should be different (e.g. blocked instead of activated). In any case where the panels cannot agree, the result is always to reject the Assigned Code Point in the zone repertoire, or to reject the code point variant rule. This is in keeping with the Conservatism Principle.

B.3.2.2 Communication between panels

The panels may discuss points of disagreement (or probable disagreement) at any time, as formally or informally as they see fit, provided that all such communications are treated as being publicly available. A useful mechanism might be a mailing list for the secondary panel, with a public archive. Interaction between the secondary and primary panels should be as formal as necessary for productive work, but need not be more formal. Our intention is that the final label generation rules be clearly the product of collaboration among diverse communities (and members of those communities), rather than being the product of the secondary panel alone.

B.3.2.3 Decision of the secondary panel is atomic

The secondary panel's evaluation of a given primary panel's output is atomic: either it accepts a proposal completely, or it rejects it completely. That is, the secondary panel is free to reject a primary panel's recommendations, but it is not free to amend the details of the primary panel's proposal. It may, however, return a primary panel's proposal with a suggestion about what would change the opinion of the secondary panel. Most importantly, the secondary panel is required to provide detailed reasoning for its rejection in every case. The secondary panel's decision cannot be appealed, but the primary panel is free to alter the proposal and submit it again.

B.3.2.4 Output from secondary panel

The secondary panel creates a set of recommended label generation rules that includes the union of all the approved proposals from the primary panels.

When the secondary panel has created such a set, it is posted for public comment using the prevailing ICANN procedures. If any of the proposals from a primary panel are under dispute, they are to be excluded from public comment. Instead, the recommendations must include a note indicating the outstanding dispute and the measures being undertaken to resolve it.

If the conflict is resolved during the public comment period, the public comment period is immediately cancelled (notwithstanding any current ICANN procedures) with an announcement that the dispute is resolved, and new recommendations are pending. If the conflict is not resolved during the public comment period, the secondary panel receives and reviews the public comment. If as a result it makes alterations, the output is treated as a new secondary panel output. When the secondary panel makes no more alterations due to public comment, the resulting label generation rules become the new label generation rules for the root zone.

B.4. What Panels to Create, When, and for What Scope

The implementation of the procedures in this document will require the creation of panels. Because the secondary panel reviews the primary panels at time of the latter's creation, the secondary panel should be created first. The secondary panel is also tasked with establishing the overall starting point repertoire.

A good guide for creating the initial set of primary panels would be to use the U-labels either that are already in the root zone or for which there is a pending application. The mere existence of an application (or even a delegation) using a code point is not sufficient reason to allow the code point in question; but it is positive evidence of some level of desire to use this code point, and by extension, to use the script to which this code point belongs.

Primary panels are chartered to work at "natural boundaries" for their task. Primary panels shall not be chartered in such a way as just to permit a small number of already requested Assigned Code Points. Instead, primary panels work on a writing system's alphabet, or all the code points from a single script where that script is used by several different languages, and so on. The exact boundaries of the scope of each panel cannot be stated in abstract (since cases are sensitive to the vagaries of writing systems). An initial list of the recommended primary panels is in section B.6.3.

Among the initial primary panels a special panel of experts may be chartered to deal with the "easy cases": single, well-defined scripts that are used for exactly one language. For a proposed scope for such a panel, see section B.6.3.6.

Once the procedure has produced the first root label generation rules, it is used for future iterations of the rules. New primary panels may be chartered whenever there is reasonable evidence of interest on the part of some language or writing system community, subject to the diversity requirements outlined in section B.2.1. One important piece of evidence (which is yet neither necessary nor sufficient) in favor of creating a primary panel would be the desire of someone to apply for a TLD using code points not already in the repertoire for the root zone. In case there is sufficient interest in an as yet unconsidered (or excluded) writing that uses an already examined script, the primary panel for that script would be asked to reconvene.

B.4.1. The secondary panel is more general

The two-panel approach depends on the idea that the two panels have different purposes. The primary panel is expected to be specialist. If it does not have deep and wide expertise in the script(s) it is considering, it will need such expertise. The secondary panel includes specialists in Unicode and writing systems, but they are to be linguistic generalists: arguments that depend on the secondary panel understanding particular points about the language or script in question should probably be rejected as insufficiently generic for use in the root zone.

B.5. Starting points for the panels

B.5.1. Panels start with the latest version of Unicode

It is possible that, at the time the work begins, there will be available a version of Unicode that has not yet been evaluated for use with IDNA2008. Panels will start with the latest version of Unicode anyway. This is consistent with the Longevity Principle: the ultimate label generation rules should be stable for the new version of Unicode too, and the properties of any Assigned Code Point must be stable as compared to the previous Unicode version, or else the code point in question violates the Stability Principle. Assigned Code Points new to Unicode, however, and those that have had altered properties in the latest version, should perhaps be left

out of the repertoire in any case, because of the Usability and Conservatism Principles.

In assessing the stability of a character's identity and usage, a recent change in its Unicode character properties can be an indicator of lack of stability. However, Unicode defines over one hundred character properties. Some are normative, some informative and some are provisional. A change in some property assignments, such as Script_Extensions, would normally be less of an indicator about a change in a character's identity or use, but rather the result of details of its usage having become better known over time. It is therefore incumbent on the panels to use judgment in evaluating the stability of characters.

To ensure that all primary panels start off with characters that meet certain minimal requirements for consideration as part of this process, the secondary panel will initiate the process by defining the set of code points that it considers to fulfill certain minimal criteria to be eligible as part of the root repertoire, pending further review during the two-stage process. The final repertoire is then expected to be a subset of this initial set. Primary panels and secondary panel may further exclude code points, and because of the Inclusion principle, only those code points that some primary panel requests out of this overall repertoire would actually be eligible for addition to the final repertoire.

B.5.2. Relationship to existing IDN tables

The existence of IDN tables (which are themselves in part expressions of a code point repertoire for some zone) registered with IANA is inadequate for the purposes of establishing or maintaining the root zone repertoire. This is because of the Usability Principle, and the fact that the root zone's user population is the entire Internet population. Assumptions about user populations that might be appropriate for particular zones (especially those of ccTLDs) can be mistaken in the context of every language on the planet.

At the same time, the existing repertoires may be useful starting points in two ways. First, if an Assigned Code Point is available in any repertoire, that may constitute weak evidence of a need for that Assigned Code Point. Second, if an Assigned Code Point is available in several repertoires, it may be evidence of need for that Assigned Code Point, or evidence of contentious use of the Assigned Code Point, or both.

B.5.3. Transitivity and symmetry of rules

In order to meet the Simplicity Principle, code point variant rules need to be symmetric and transitive. That is, if the code point or series of code points V_1 has a variant rule that produces the code point or series of code points V_2, then in the label generation rulesV_2 also has a variant rule that produces V_1. Further, if V_2 has a variant rule that produces the code point or series of code points V_3, then V_1 must also have a variant rule that produces V_3. This requirement may on occasion

produce labels that would be incorrect, and may also deliver variants to the exclusion of other possibly useful labels. It is nevertheless appropriate in the root zone, where the goal is not to maximize the number of possible labels but to minimize the confusion possible in a shared environment supporting heterogeneous linguistic communities.

B.5.4. Relationship to Unicode properties

Every assigned Unicode code point has exactly one script property, and it may be tempting to use that to restrict the Assigned Code Points available to be considered by any first-pass panel, and to regulate what other Assigned Code Points are permitted in any variant rule. Unfortunately, such an approach suffers from four defects:

1. Many (perhaps most) languages use Assigned Code Points from more than one script, particularly if the Inherited and Common scripts are considered independently.
2. Conversely, if Common and Inherited are simply included in every other script, the category is too broad. There are, for example, Assigned Code Points in the Common script that are not used with Latin, but U+002D HYPHEN MINUS is in the Common script.
3. Many languages use Assigned Code Points from the Latin script, particularly in a computing context, even though they are normally written using a different script.
4. Even if a first-pass panel is restricted to a single script, the second-pass panel will need to consider cross-script cases (e.g. U+0061 LATIN SMALL LETTER A vs. U+03B1 GREEK SMALL LETTER ALPHA vs. U+0430 CYRILLIC SMALL LETTER A). While resolving string-confusability issues is beyond the scope of this project, the second-pass panel will need to take into consideration the consequences of the label generation rules for the Usability and Conservatism Principles.

Accordingly, it is not possible *a priori* to limit a first-pass panel to only one script property, and it is even less desirable so to limit the second-pass panel.

The most recent version of Unicode contains the Script_Extensions property. One of the things it is intended to do is to narrow the perhaps overly broad Common and Inherited scripts. It appears that Script_Extensions will be a useful tool with which to restrict the scope of work for Primary panels.

The first task of the secondary panel would be the definition of the maximal set of code points that it considers to fulfill certain minimal criteria to be eligible for consideration in the remainder of the process. The code points in this set would meet the following conditions:

- Assigned in the latest version of the Unicode Standard

- Not explicitly excluded by IDNA2008
- Not restricted for identifiers in Table 1 of UTS#39
- Not used for writing an excluded script

Excluded scripts are those that, according to the secondary panel, do not have a living language community. This would pre-empt inclusion of the code points having those script properties from ever being included in the zone repertoire. In making its decision, the secondary panel must consider evidence of actual speakers and writers of a language as part of its evidence. Because scripts that are excluded on the basis of no living language community do not have to be considered when reviewing the output of primary panels, such scripts are effectively permanently barred from future consideration. Therefore, the secondary panel will be conservative in excluding a script on the basis of it having no living language community, and if there is a doubt the secondary panel must not exclude the script. Writing systems used by very small numbers of people ("endangered languages") do not meet this test. Writing systems used by now extinct languages (e.g. Linear B) could be excluded. See section B.6.3.1 for our recommendations of initially excluded scripts.

In section 3.1, UTS#39 includes a mechanism for evaluating Assigned Code Points to determine whether they are appropriate for use in identifiers. This determination is based in part on whether a code point is part of a script not used for writing a living language, or a script that is of limited use, or otherwise not yet widely used, as defined in UAX#31, Tables 4 through 7. A listing of the code points thus restricted can be found at http://www.unicode.org/Public/security/latest/xidmodifications.txt.

Some of the scripts identified in UAX#31 might be eligible for the root after all. This would be an area for judgment by the secondary panel.

Primary panels must not include in their proposed repertoires any assigned code point that is not included in the maximal repertoire defined by the secondary panel.

B.5.5. Distinguishing among states resulting from variants

IIR explores at length the difference between variants that are intended to go into the zone, and variants that, because of the existence of some other label, must not go into the zone. See Section 5 of IIR. We accept those states as a foundation for the present work.

The secondary panel may deliver a rule that has one of three results: allocation, withholding, or blocking.

B.5.5.1 Allocation

An allocation rule says that once the variant label is generated, that variant label is allocated to the applicant for the original label. The allocated label is then subject to any activation restrictions that might be appropriate under prevailing ICANN policies or agreements (or both); but in principle, any label that is allocated may be delegated, according to the wishes of the party to whom the label is allocated.

B.5.5.2 Withholding

A withholding rule says that, once a variant label is generated, that variant label is withheld from being allocated to anyone pending the removal of the first label from allocation, or the satisfaction of some other condition. A label that is withheld might, in principle, become eligible for allocation at some time in the future. It is not clear what the consequences of such a change would be, and it seems likely that some investigation of the consequences to users would be prudent.

B.5.5.3 Blocking

A blocking rule says that a particular label must not be allocated to anyone under any circumstances. We may distinguish between two types of blocking. The first is simply a consequence of the non-inclusion of an Assigned Code Point in the zone repertoire. By definition, any U-label that contains a code point not in the zone repertoire is blocked. This could change if a subsequent repertoire expands to include the formerly excluded code point. The second type of blocking is an explicit decision to block the resulting label. A change to such a rule would require study, could only be undertaken case by case, and would suggest serious problems with this procedure in light of the Conservatism Principle.

B.6. Other considerations

B.6.1. How early can we have some label generation rules?

Some communities have grappled with variant issues already, and they may feel they are "ready to go". Some have argued that it would be unfair to make those communities wait until everyone else in the world is ready. There is some merit to this position, since if we took seriously the requirement to wait until *everyone* is ready, we might never be able to act: we would have to wait until we were sure that the encoding of every possible writing system was complete.

The secondary panel may deliver a repertoire before waiting for all first-pass panels to complete, *provided that* it has strong reason to believe that there will be no overlap between the Unicode code point range it is delivering, and the work of an existing (or likely prospective) first-pass panel. There are two cases where this appears to be likely.

The first is for zone repertoires restricted to one script; which script is unrelated to any other script, and is used for just one writing system – what we might call an "isolated" script. A candidate zone repertoire of this first case may contain only Assigned Code Points used for one language (or set of closely related languages), and never used for anything else.

The second (perhaps more common) case is for zone repertoires that may be built from more than one script, but where a single primary panel that has, in the opinion of the secondary panel, considered the issues for all the potential users of the included Assigned Code Points. For the second case, the secondary panel must be all but certain that there are no possible uses of the Assigned Code Points included in the rules that have not been considered by the primary panel. For practical purposes, this will restrict "early ruling" to code points that are used in only a few writing systems.

B.6.2. Panels, Conservatism, and the Limits of Knowledge

In all matters the secondary panel's judgment is to be governed by the Conservatism and Stability Principles. If the secondary panel delivers a repertoire while there is still work to be done on other parts of Unicode, as is inevitable, we expect subsequent iterations of the repertoire. We expect those iterations to increase the size of the repertoire, *and not* to remove any code point or change any code point variant rules to reflect the new inclusions.

If an iteration of the process causes a subsequent repertoire to remove a code point that was in an earlier repertoire or to change an existing variant rule, all operation of the procedure must halt. A review of the process must ensue to determine whether it is effective at following the principles outlined in Section A.3, and whether it is possible (and how) to add additional checks to the procedure to avoid recurrence of similar failures. Because it is impossible to state in advance what the failure might be (since if we knew, we could write rules to avoid it), the ICANN Board will determine the nature and scope of the review, and will appoint the reviewers. If such halts are called frequently, that is a reason to believe that this procedure does not work, in which case a new procedure will be needed. One effect of the overarching Conservatism Principle should be that these events will not happen frequently; but given the procedure's reliance on human judgment, it may be necessary to tolerate errors from time to time.

B.6.3. Additional considerations

This section should be read as advice or a suggestion, but not as normative. In what follows, we discuss some particular cases where we think panels should be created that span more than one writing system, suggest the initial list of panels, and suggest some scripts that should be excluded from the start. In our view, some panels with a broader mandate are needed to arrive at recommendations that satisfy the Usability and Stability principles in particular. However, whether such

panels can be created depends on cooperation by the relevant linguistic communities.

B.6.3.1 Initial exclusions

Some scripts with characters that are (or would be) permitted under IDNA2008 have been identified as of primarily or exclusively historical use, meaning that at present we judge it to be unlikely that they would attract a meaningful audience in terms of a user community in the context of the root. This could warrant exclusion of code points having those script properties from being included in the root zone repertoire.

Scripts identified as most likely belonging to this class are Avestan, Brahmi, Carian, Coptic, Cuneiform, Cypriot, Deseret, Egyptian Hieroglyphs, Glagolitic, Gothic, Imperial Aramaic, Inscriptional Pahlavi, Kharoshthi, Linear B, Lycian, Lydian, Mandaic, Meroitic Cursive, Meroitic Hieroglyphs, Ogham, Old Italic, Old Persian, Old South Arabian, Old Turkic, Parthian, Phags-pa, Phoenician, Runic, Samaritan, Shavian, Tagalog, and Ugaritic.

Other scripts currently under ballot for inclusion in the Unicode standard, and which would in due course most likely be members of this class, are Caucasian Albanian, Duployan, Elbasan, Khudawadi, Linear A, Mahajani, Manichaean, Modi, Nabataean, Old Hungarian, Old North Arabian, Old Permic, Palmyrene, Pau Cin Hai, Psalter Pahlavi, Tangut, and Tirhuta.

B.6.3.2 Han and related

We believe it would be a good decision to convene a single primary panel to treat Han and, at the same time, writing systems used in conjunction with the Han script. In effect, this would constitute a “CJK” panel. This is in keeping with the Chinese Variant Issues Program report [ChineseVIP].

B.6.3.3 Cyrillic, Greek, and Latin

Because of the shared history of Latin, Greek, and Cyrillic scripts, it seems prudent that the secondary panel be required to have complete input from primary panels for each in order to make any determination.

There have been suggestions that every script may need to mix with at least part of Latin; this request needs careful examination.

It is, however, unrealistic to expect that the primary panels will be able to consider every writing system that is based on Latin or Cyrillic. These scripts are used in too wide a variety of languages to make this feasible.

B.6.3.4 Brahmi-derived scripts

The variant issues project developed a report only on Devanāgarī [DevanagariVIP]. The resulting report, however, hinted at issues that were common to Devanāgarī and other Brahmi-derived scripts. Therefore, it seems prudent that the secondary panel attempt to deal with all Brahmi-derived scripts at the same time, even if some of them have not attracted enough interest to create a primary panel. In addition, it seems that it would be wise to deal with as many Brahmi-derived scripts as possible in a single primary panel.

B.6.3.5 Arabic

The Arabic issues report [ArabicVIP] made plain that it did not have adequate expertise to cover all the different uses of Arabic when reporting. For the purposes of the primary panel, it will be extremely important to address those gaps when proposing the Arabic portion of the zone repertoire and associated code point variant rules.

B.6.3.6 The "easy cases"

We have identified a certain number of scripts, with small repertoires and used by one or by only a few languages, as "easy cases": at present we judge them to be relatively unproblematic, and they may not need to have very large primary panels or protracted discussion. (This does not mean that they would not require panel processing; it is just recognized that the discussions about their repertoire are unlikely to be complex or problematic.)

The scripts identified as most likely belonging to this class are Armenian, Georgian, and Thaana.

C. How the proposal aligns with the Principles

In general, the panels' deliberations are to be guided by the principles listed in Section A.3.5. Below we include some specific remarks on the ways the proposal achieves this.

C.1. Longevity Principle

Assuming the panels are doing their work, the Longevity Principle should be enforced by both the primary and secondary panel. The panels are supposed to begin using the latest version of Unicode, but also to take into consideration the stability of Unicode character properties. If the panels both fail to behave this way, then there is a risk either that code points will be permitted for allocation in the root zone that do not work with multiple versions of Unicode, or that code point

substitution rules will be adopted that work well in peculiar contexts, but that will work poorly in other (perhaps future) contexts.

C.2. Usability Principle

The Usability Principle aims at ensuring that the Allocated Code Points included in the zone repertoire are useful as elements in unique identifiers. To the extent that a code point is confusing to the user population – either by accident or else by way of malicious use – use of the code point fails to adhere to the Usability Principle in that context.

The secondary panel, especially, is responsible to ensure adherence to the Usability Principle. It is explicitly charged with considering the entire user population, which is everyone on the Internet.

C.3. Inclusion Principle

The proposal is an example of the Inclusion Principle in action, since every rule or code point is excluded until reviewed and explicitly included.

C.4. Simplicity Principle

Part of the point of the secondary panel is that it performs a check of the Simplicity Principle. The secondary panel cannot possibly include experts in every language and script, but the members have general knowledge of Unicode, IDNA, DNS, or all of the above. If any member of the secondary panel cannot understand the rationale for inclusion of some rule, then that member will not support the rule, and it will not proceed. This is the purpose of the unanimity requirement for the secondary panel.

C.5. Predictability Principle

The proposal follows the Predictability Principle in much the same way it follows the Simplicity Principle: if the secondary panel does not immediately agree with the recommendations of the primary panel, or if members of the secondary panel disagree with each other, that is a good reason to suppose that the rule in question is not really predictable.

C.6. Stability Principle

Especially in the case of the root zone, the Stability Principle is less a matter of guidance and more a statement of fact. The proposed procedure attempts to minimize the possibility that an Assigned Code Point or any other label generation rule will be permitted for the root zone without that rule having been considered as carefully as possible for any negative consequences. If there is a failure such that

the secondary panel determines that a previously-active rule needs to be removed, this proposal requires that the procedures themselves be subject to review.

C.7. Letter Principle

The secondary panel is required to follow the Letter Principle in its deliberations.

C.8. Conservatism Principle

The proposal is consistent with the Conservatism Principle in two ways. First and most important, because the secondary panel is supposed to reject anything it does not positively think is safe, the Conservatism Principle is built in to the secondary panel's criteria. Second, in the event of disagreement between the primary and secondary panels, the proposed rule that is the point of disagreement is automatically excluded from the root label generation rules.

D. Evaluation of this Proposal Against the "Parameters"

Section A.3.6 introduced the four independent parameters that can be used in evaluating the proposed process for label generation rules.

- Comprehensiveness
- Expertise
- Qualification
- Centralization

The remainder of this section will describe these parameters in detail and use them to evaluate the proposed process.

For each parameter, we will examine their possible extreme values and consider their consequences on the IABCP principles outlined in Section A.3.5.

After that brief analysis, we will use these parameters to evaluate the proposed procedure for creating and maintaining the label generation rules

D.1. Overview of the Parameters

D.1.1. Comprehensiveness

The comprehensiveness parameter describes the extent to which all of Unicode is being considered. The maximal setting corresponds to the requirement that every code point in Unicode be evaluated before proceeding. The minimal setting would require review only of code points actually requested for allocation in the root zone.

Because Unicode changes from version to version, it is actually impossible to consider "all of Unicode": a future release will introduce new code point assignments that may be permitted under the IDNA2008 specification. Those code points will by definition not have been considered by a panel considering an earlier Unicode version. While it would be possible in principle to consider every code point in some version of Unicode, under the Inclusion Principle any code point not explicitly included would be excluded. And, since unassigned Unicode code points are DISALLOWED under IDNA2008, when the panels are considering the older version of the Unicode character repertoire they are, by definition, not including code points that will be assigned in a later version of Unicode.

It is not enough simply to investigate code points. A comprehensive analysis requires one also to take into account the way each language or writing system uses its repertoire of code points.

Consequences of requiring maximal comprehensiveness

The consequence of requiring maximal comprehensiveness is mostly procedural: considering every code point would take a very long time, and might never complete. This is not a risk in terms of the IABCP Principles, but ICANN would face considerable pressure to do something in the meantime, and if it did that would almost certainly violate the Conservatism Principle. In addition, it is possible that, when investigating all of Unicode, those performing the investigation will be tempted to rule in favor of including a code point or associated rule they do not, or do not fully, understand. This would violate the Conservatism and Inclusion Principles.

It would be possible to reduce the above risks by considering a smaller subset of Unicode – that is, by requiring less than maximal comprehensiveness. The consequence of that would be the risk of later additions.

Consequences of accepting minimal comprehensiveness

Minimal comprehensiveness introduces a high risk of subsequent violations of the Stability and Usability Principles. If evaluations are made only for code points as they are requested, then a later request could introduce factors that were not under consideration in an earlier evaluation. Because a later request is likely to include code points not previously requested, later evaluations will have to expand the repertoire. If those code points were not considered previously, then there is some risk that there will be changes to the rules (and by definition, there will be changes to the repertoire). Those changes could include new rules that introduce a conflict with older rules, making a formerly acceptable code point into one that is unacceptable.

That would be a violation of the Stability principle.

D.1.2. Expertise

The Expertise parameter reflects who is involved in establishing the repertoire and rules. IIR includes under this description both the question of the degree of expertise and the degree of centralization in the development in the rules; in the present case, we are distinguishing between these parameters; see section D.1.4 for discussion of centralization. Requiring maximal expertise would place the entire burden for development on experts in the subject area. Experts would be needed in all the relevant topics, including at least Unicode, software internationalization, IDNA, and DNS; for all of these, expertise in both protocols and operations is required. A minimal requirement, in contrast, would accept a rules definition from anybody, regardless of their knowledge of the script or protocols in question.

Consequences of requiring maximal expertise

It is hard to see how requiring maximal expertise could result in any violation of the Principles, but such a setting could be practically unsustainable. Because a panel of experts would have to undertake all development itself, there is the potential that it could take a long time. Moreover, if those desiring to register IDNs in the root zone are not included in the development of the rules, then there is a considerable risk that they will object to the experts' judgment when it is rendered. If a language purist were to be part of the panel, it is possible that such a participant could effectively prevent an outcome he or she did not like.

Consequences of accepting minimal expertise

Since accepting the minimum would require no expertise at all for the establishment of rules, there is every reason to suppose that any resulting rules would violate the Conservatism Principle whenever that principle does not yield a result someone wants. It would also likely produce divergent or inconsistent rules, thereby violating the Simplicity and Predictability principles.

D.1.3. Qualification

The Qualification parameter reflects the extent to which code points can be restricted *a priori* for inclusion, or the rules about them determined at least partly according to properties of those code points. (It might also be called the Automaticity parameter; here we follow the IIR and use the name "Qualification".) A maximal setting for this parameter would result from using something like the Unicode script property, and establishing rules that only permit labels made of code points all with the same script property. To prevent confusion, also, code points with the script property Common or Inherited would be automatically disqualified. A minimal setting of this parameter would correspond to using no property of code points in evaluation, relying instead on arbitrary combinations within the bounds of IDNA2008.

Consequences of requiring maximal qualification

Because the Unicode script property does not map perfectly to any writing system – especially in the context of the DNS – strictly qualifying characters is likely to be very surprising for at least for some classes of user, not least because of the exclusion of Common and Inherited code points. Moreover, a single writing system as used by one language may contain rules inconsistent with other languages using the same writing system; this violates Usability, Simplicity, and Predictability. Maximal qualification, therefore, seems at once too broad and too narrow.

Consequences of accepting minimal qualification

Applying minimal or no qualification runs the risk of violating the Usability Principle, particularly in respect of abuses. In addition, the minimal setting seems likely to permit violations of the Letter Principle, though it is hard to see what formal property could be used to ensure that the Letter Principle will be followed without also running into the problems outlined in IABCP (particularly Section 2).

D.1.4. Centralization

The centralization parameter describes the extent to which rules are made by a single group of people. A maximal setting would mean that all rules are proposed, considered, and set by a central committee. Accepting a minimal setting would allow rules defined by anybody at all. The case of minimal centralization would require some, possibly centralized, mechanism for conflict resolution.

Consequences of maximal centralization

This may not violate any of the Principles, but might be politically unacceptable because of an appearance that it is not sensitive to community concerns. The required expertise is also so diverse that it is unlikely that it can be collected effectively in a single body.

Consequences of accepting minimal centralization

A free-for-all approach runs considerable risk of violating the Usability, Conservatism, Simplicity, and Stability Principles, and might violate the Predictability Principle as well.

The first issue with minimal centralization is that different types of users will almost certainly submit rules for different ranges of Unicode code points. There is every reason to believe that different people will treat the same code point in different ways. If the different rules are not reconciled, then the total set of rules will likely violate the Usability and Simplicity Principles, as well as the Predictability Principle. In any case, such unreconciled rules violate Conservatism and Predictability,

because the total set of rules might not be internally consistent. If the rules are reconciled, then Conservatism is violated, since a later addition of new rules is likely to violate the Stability Principle.

It seems plain that, even with minimal centralization, the rules still need to be reconciled. By definition, at the minimal setting there is no group of experts to reconcile the rules, so the only mechanisms for reconciliation are first come, first served; or a secondary rule that, in the event of conflict, both conflicting rules are removed from the rule set. The first come, first served mechanism is in clear violation of the Usability, Predictability, and Simplicity Principles: it does not address the entire root zone user population, and the only way to make the rule predictable and easily understood is to know the order in which requests arrived. On the other hand, the second mechanism, of denying both conflicting rules, provides a simple method for someone to prevent all IDN labels in the root zone, by submitting one rule designed to conflict with any other rule that is submitted.

D.2. This procedure and the various parameters

The boundaries outlined above suggest that a successful procedure for generating label generation rules will correspond to a setting of the four parameters that minimizes the risks of complete paralysis, while yet minimizing the labels that are permitted. The overarching Conservatism Principle suggests that allowing the smallest number of possible labels is desirable; and, given the user population of the root zone (i.e. everyone who ever uses the Internet), the Usability Principle, with its focus on positive recognition and limits on possible misuse of labels leads to the same conclusion. The Longevity, Stability, Simplicity, and Letter Principles all militate in the same direction.

D.2.1. Comprehensiveness

The proposed process recognizes that it is effectively impossible to review even a single version of the Unicode Standard in a comprehensive manner. The reason goes beyond the sheer number of code points – itself a daunting problem – and extends to the need to review the use of these code points by each and every contemporary writing system. For many of these writing systems, it is effectively impossible to access the required information. Moreover, the Unicode Standard itself has not completed its task of supporting all such writing systems, which is part of why it is being updated regularly as additional information becomes available.

The proposal recognizes the essential additive nature of the process and the requirement to allow for a similar additive nature in terms both of repertoire and variant rules. At the same time, the initial phase of the work should attempt to deal with the repertoires and writing system for at least the major scripts and user communities.

The process does not require a comprehensive consideration of everything in Unicode, because it relies on the establishment of primary panels to tackle subsets of Unicode. Those primary panels are based on communities of interest, and we presume that some portions of Unicode will not be interesting. Indeed, some portions of Unicode are simply marked as not eligible, on the grounds that there is no living language community that uses those ranges of Assigned Code Points.

As a brake on exuberance, the secondary panel is charged with ensuring that the final set of label generation rules takes into account not only then-current, but likely future uses of Unicode in the root zone; this includes effects from code points or rules not actively in use or under consideration, but that might be requested in future.

We may say, then, that the process offers a moderate level of comprehensiveness. It attempts to offer progress – particularly for those scripts where the issues are well understood – while yet constraining the label generation rules so that the Stability and Usability Principles are followed.

D.2.2. Expertise

Because of the two-panel structure, the procedure relies upon a high degree of expertise without confining itself only to experts. Initial development is undertaken by panels that are neither necessarily expert in any particular area, nor in the whole of Unicode. The secondary panel, on the other hand, has a responsibility for total review and is ultimately responsible for the label generation rules as deployed in the root zone.

The proposal recognizes that a shared resource, like the root zone, requires cross-script expertise, but that each script and writing system will bring its own issues. Instead of requiring a single expertise level, the proposal provides for different levels and types of expertise at each level of panel.

The expertise parameter for the primary panels could be said to be at a medium or moderate level, while for the secondary panel it would be at a high value.

This approach permits timely progress and ensures that opinions other than those of the experts are taken into consideration during development. At the same time, requiring a final ruling by disinterested experts increases the probability of following the Conservatism Principle.

D.2.3. Qualification

The proposal opts for pre-defining qualification largely in the negative sense. The first is a consequence of IDNA2008, which excludes many code points. The second would disqualify any code points and scripts not used for everyday writing. This includes dead scripts, as well as specific code points used for dead languages,

specialized uses such as phonetics, and the like. Finally, the secondary panel will only permit in the zone repertoire those code points normally used to write words. There is no single Unicode property that covers any of these restrictions, although the repertoire for historic scripts can be derived from the script property.

The procedure does not impose a formal link to any Unicode property for purposes of qualification, in an attempt to permit the primary panels to select those parts of Unicode that they believe to be the best fit. Overall, this corresponds to an intermediate value for the Qualification parameter. Having whittled down the eligible possible characters, the precise Unicode properties or other information to be used for the qualification of characters is left to the primary panels to sort out, with the important limitation that code points not suited for use in identifiers are not allowed.

D.2.4. Centralization

The procedure attempts to strike a balance between the control and consistency that may come from having a central authority, and the political and technical realities of needing a single, internally consistent set of rules to govern the root zone.

The proposed process uses two panels: the first to allow broader community input for each part of the repertoire, and the second to provide a centralized body of experts able to resolve conflicts and to represent the needs of the root zone as a whole. The proposal as a whole is characterized by a moderate level of centralization, while allowing for highly centralized reconciliation of the primary panel output.

The requirement for consensus between the primary and secondary panels is intended to ensure that the reconciliation process will give results in line with the Conservatism principle.

E. References

[ArabicVIP] Hussain, S, *et al.* "Internationalized Domain Names Variant Issues Project Arabic Case Study Team Issues Report". (Marina del Rey, California: ICANN, October 2011). http://archive.icann.org/en/topics/new-gtlds/arabic-vip-issues-report-07oct11-en.pdf.

[ChineseVIP] Lee, X. *et al.*, "Report on Chinese Variants in Internationalized Top-Level Domains". (Marina del Rey, California: ICANN, October 2011). http://archive.icann.org/en/topics/new-gtlds/chinese-vip-issues-report-03oct11-en.pdf.

[DevanagariVIP] Govind, *et al,,* "Devanāgarī VIP Team Issues Report". (Marina del Rey, California: ICANN, October 2011). http://archive.icann.org/en/topics/new-gtlds/devanagari-vip-issues-report-03oct11-en.pdf

[LatinVIP] Frakes, J, *et al.,* "Considerations in the use of the Latin script in variant internationalized top-level domains: Final report of the ICANN VIP Study Group for the Latin script". (Marina del Rey, California: ICANN, October 2011). http://archive.icann.org/en/topics/new-gtlds/latin-vip-issues-report-07oct11-en.pdf.

[IABCP] Sullivan, A., Thaler, D. , Klensin, J. , and O. Kolkman, "Principles for Unicode Code Point Inclusion in Labels in the DNS." draft-iab-dns-zone-codepoint-pples-00.txt. Work in progress. Available at http://wiki.tools.ietf.org/html/draft-iab-dns-zone-codepoint-pples-00. Visited 2012-09-21.

[IIR] Internet Corporation for Assigned Names and Numbers, "The IDN Variant Issues Project: A Study of Issues Related to Management of IDN Variant TLDs (Integrated Issues Report)." (Marina del Rey, California: ICANN, February, 2012). http://www.icann.org/en/topics/idn/idn-vip-integrated-issues-final-clean-20feb12-en.pdf.

[ISO15924] *Codes for the representation of names of scripts*, ISO 15924:2004. Available from http://www.unicode.org/iso15924/. Visited 2012-09-21.

[RFC3743] Konishi, K., Huang, K., Qian, H., and Y. Ko, "Joint Engineering Team (JET) Guidelines for Internationalized Domain Names (IDN) Registration and Administration for Chinese, Japanese, and Korean", RFC 3743, April 2004.

[RFC4290] Klensin, J., "Suggested Practices for Registration of Internationalized Domain Names (IDN)", RFC 4290, December 2005.

[RFC5646] Phillips, A. and M. Davis, Eds., "Tags for Identifying Languages", RFC 5646, BCP 47, September 2009.

[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.

[RFC5894] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Background, Explanation, and Rationale", RFC 5894, August 2010.

[RFC5895] Resnick, P. and P. Hoffman, "Mapping Characters for Internationalized Domain Names in Applications (IDNA) 2008", RFC 5895, September 2010.

[UAX29] UAX #29: *Unicode Text Segmentation.* Available from http://www.unicode.org/reports/tr29/. Visited 2012-09-21.

[UAX31] UAX #31: *Unicode Identifier and Pattern Syntax.* Available from http://www.unicode.org/reports/tr31/. Visited 2012-09-21.

[Unicode61] The Unicode Consortium. The Unicode Standard, Version 6.1.0, defined by: "The Unicode Standard, Version 6.1", (Mountain View, CA: The Unicode Consortium, 2012. ISBN 978-1-936213-02-3). http://www.unicode.org/versions/Unicode6.1.0/.

[UTS39] UTS#39: *Unicode Security Mechanisms*. Available from http://www.unicode.org/reports/tr39/. Visited 2012-09-21.

Appendix A Example of a functioning label generation rules with variants

Let us suppose that a Latin panel, going against the advice in the Latin variant issues report [LatinVIP], determined that it needed to produce variants for Latin characters. Suppose further that the Latin panel determined that it would simply be too hard to produce language-specific rules, so it established a single tag for all of Latin. The tag in this case is "und-latn-TBD-root". This is an imaginary example, is probably wrong in the details, and is not intended to guide any future primary or secondary panel in any deliberation.

Suppose that the primary panel determines that, to be useful, the variant relationship in Latin is from a single "base" character to every "decorated" character in the script. In effect, every character in the ASCII alphabet has a variant with that character and every possible diacritic used with that character.

So, for instance, the character U+0073 LATIN SMALL LETTER S (s) might be in the repertoire. Its code point substitution rules might include U+015B LATIN SMALL LETTER S WITH ACUTE (ś), U+015D LATIN SMALL LETTER S WITH CIRCUMFLEX (ŝ), U+0161 LATIN SMALL LETTER S WITH CARON (š), and so on. Similarly, the character U+0064 LATIN SMALL LETTER E (e) might be in the repertoire. Its code point substitution rules might include U+00E8 LATIN SMALL LETTER E WITH GRAVE (è), U+00EB LATIN SMALL LETTER E WITH DIAERESIS (ë), and U+1EBB LATIN SMALL LETTER E WITH HOOK ABOVE (ẻ).

Thus, when someone applied for the label "test", the label generation rules would also generate variants: teśt, teŝt, tešt, tẻst, tẻšt …

In this particular case, it seems unlikely that any of the code points would lead to a blocked label, so every one of these would be allocated. Whether any of them was delegated would be a separate matter to be determined under prevailing ICANN policy at the time. Alternatively, perhaps the Latin code point variant rules say that, if any one of the menu of characters is delegated, then all the others must be blocked. In that case, the label "test" would be allocated, and all the other variants blocked so that nobody else could successfully apply for them.

Importantly, for reasons of symmetry, if the application was for "teśt", the list of variants would be the same. In case all variants were to be blocked, it would cause the (non-IDN) label "test" to be blocked. This illustrates the way that the IDN repertoire for the zone has implications for traditional LDH-labels (and conversely).

Appendix B Examples of structurally invalid strings

There are strings that are structurally invalid, but that are possible to type. For instance, it is possible to put together a series of combining marks that would be IDNA2008 PVALID but that should not be permitted as labels. For instance, the character U+0300 COMBINING GRAVE ACCENT is PVALID. So, it is possible to construct a PVALID string U+0300 U+0300 U+0300 U+0065 (it looks like this: ◌̀◌̀◌̀e). Such a string is not structurally valid, however: it is a string that begins with three combining marks.

The purpose of the final whole-label evaluation, where it is in place, is to prevent cases like these from being possible labels in the root zone.