IDN Variant TLDs – Cyrillic Script Issues

1. Introduction

The ICANN IDN Variant TLDs Issues Project is investigating issues around IDN variants and the global DNS, with particular attention to the issues for top-level Internationalized Domain Names using IDNA2008 (and, therefore, Unicode). Case study teams for six individual scripts were asked to investigate the set of issues that need to be resolved to facilitate a good user experience for IDN variant TLDs. The Project Plan can be found at http://www.icann.org/en/topics/new-gtlds/idn-variant-tlds-delegation-20apr11-en.pdf.

The following are within the scope of work for the team:

1. Identify appropriate terminology for the various concepts and requirements, ensuring such terms are accurate and vetted with appropriate technical and linguistic communities and are used consistently throughout the project to improve the dialogue among participants;

2. Identify the requirements considering (a) linguistic accuracy, (2) technical feasibility, (c) usability, (d) accessibility, and (e) security and stability.

The following items are not within the scope of the work for the team:

3. Determine the circumstances (where they exist) where certain types of IDN variant TLDs might be eligible for delegation;

4. Analyze and arrive at rules where possible, or guidelines where rules are not possible, that address the challenges of working with IDN variant TLDs outlined in task 2;

5. Arrive at rules and guidelines, both in the registry operational requirement area and the technical implementation area;

6. Determine the responsibilities of TLD operators who would be responsible for managing such delegated IDN variant TLDs;

7. Determine what kind of compliance programs may be necessary to ensure that IDN variant TLDs operate according to the arrived at rules and guidelines;

8. Identify viable and sustainable outreach mechanisms to communicate and interact with the community on the issues report.

The Cyrillic script is an alphabetic writing system that serves as the basis for many languages in Central & Eastern Europe, and North and Central Asia. A list of the languages using Cyrillic as a base script is included with this report in Appendix A.

Cyrillic also has some common or similar characters with the Greek and Latin scripts. Further background on Cyrillic can also be found in RFC 5992, Internationalized Domain Names Registration and Administration Guidelines for European Languages Using Cyrillic (http://tools.ietf.org/html/rfc5992).

Serbia, Montenegro and Federation of Bosnia and Herzegovina (including Bosnian Republika Srpska) all use both Cyrillic and Latin scripts.

The team agreed on a set of working principles in their work:

a) The contents of the DNS are about mnemonics, not about "words" or longer statements in particular languages. The fact that something can be written in a particular language, or even looked up in its dictionary, does not imply an entitlement to have that string appear in the DNS. Nevertheless, the aspiration is to implement an approach that approximates the natural language usage as nearly as possible.

b) This issues report is limited to IDN variant TLDs alone (with specific reference to Cyrillic) and may not apply to registration under subordinate zones, although the issues discussed in the report could provide gainful insights into the functioning of those subordinate zones. Thus, the report is focused on recommendations for the root. In the course of considering this set of issues, some issues that may be relevant at lower levels are also identified and discussed. The root zone cannot make use of context; therefore, it may call for different rules from what might be appropriate in a zone elsewhere in the DNS.

c) There are over 60 languages that use the Cyrillic script, as identified in Appendix A. The team included a representation from some, but not all, of those languages. Languages represented on the team include: Russian, Ukrainian, Bulgarian, Macedonian, Serbian, Bosnian and Montenegrin. While the team has been as comprehensive as possible in considering the issues present in these languages, there may be additional issues in languages not represented in the group that have not been identified in the report. The absence of considerations discussed for a particular language should not be taken as an indication that these languages are not significant. The team did not have access to experts of these languages in the writing of the report.

   The table below lists the approximate number of native speakers for the languages that are represented in the Case Study Team, with an estimated number of Cyrillic language users of 300 million[1]. The numbers below are given

[1] The corresponding figures in the 2009 edition of Ethnologue (ed. Paul Lewis, Dallas: SIL International) are: Russian 144 mil.; Ukrainian 37 mil.; Bulgarian 9 mil.; Serbian 7 mil.; Macedonian 2.1 mil.; Bosnian 2.2 mil; Montenegrin (not given: Serbian in Montenegro 0.2 mil; Bosnian in Montenegro 0.05 mil.).
In the 1989 Soviet census of Russia, the number of Russian speakers was given as slightly under 120 mil. (119,866,000),

only in order to evaluate the overall scope of the group expertise and are not suggesting in any way that good user experience is less important for speakers of other Cyrillic script-based languages.

Russian	175 mil.
Ukrainian	47 mil.
Bulgarian	12 mil.
Serbian	10 mil.
Macedonian	3 mil.
Bosnian	2 mil.
Montenegrin	0.6 mil.

d) The team's objective is to identify the issues relevant to the Cyrillic script. They are not tasked with developing solutions, and therefore this is not the focus of the report. Solving the issues identified in the project is expected to be the focus of follow-on projects by ICANN policy development, implementation plans, relevant technical work by IETF and other organizations. The key focus of the case study groups in this stage was to come up with an agreeable definition of the needs of the script users. It will be the role of the combined issues report to harmonize the differing requirements, terminology and other aspects identified by the case study groups. Areas identified as needing further work or research will be discussed in the Integration and Solutions phases of the IDN Variant Project.

e) In considering the issues, there is a need to balance the natural expectations of users with the limitations of the DNS. The DNS, and especially the root zone, are shared resources across users of all scripts, and are depended on by every user. Shared use of a single resource necessarily means that particular user communities may run into collisions with others. Problems will arise because of whole script confusables (two labels, each in a single script, that are confusable with one another. Example: "pear" in Latin and "pear" in Cyrillic.) See http://www.unicode.org/reports/tr39/#Confusable_Detection.

In this report, the team has identified examples of code points likely to cause certain problems. The absence of comment from the team about a particular character or set of characters as an example does not indicate that the group believes that there are no potential issues with it.

2. Definitions

However, 3 other (higher) estimates are given for Russian at en.wikipedia.org/wiki/Russian_language: viz G. Weber, "Top Languages", *Language Monthly*, 3: 12–18, 1997, ISSN 1369-9733: 160 mil.; World Almanac (1999) 145 mil.; CIA World Factbook 160 mil.(2005).

The Cyrillic case study team supports the Draft Definitions[2] document in general and proposes the following additional definitions:

Alternate Names:
Two names are alternates of one another just in case, for a namespace starting with one, the namespace starting with the other is isomorphic to the first, subject to the usual DNS loose consistency strictures. In the current DNS, there are 2 different techniques for this. The first is aliasing: CNAME, DNAME, and other such techniques redirect a name or a tree, effectively substituting one label for another during DNS lookup. The second is by using provisioning constraints, such that an underlying provisioning system always effects a change in all of the alternate names whenever that change is effected in one of the alternates. A fuller discussion of this topic is included for information in Appendix B.

Composite-character variants:
Abstract Characters that do not have a single assigned code point assigned, but can be represented by multiple code points.

Domain Name Blocking Policy:
Refers to a policy that has effect of certain domain names in a TLD registry becoming unavailable for allocation (for example, due to implementation of variant-related policies).

Domain Name Bundling:
Registration technique that makes multiple domain names share all registration parameters (such as creation/expiration date, associated name servers etc.) except the domain name itself. Changes to any of these registration parameters should normally take effect on all the domain names in a bundle.

Reserved name:
A name set aside for a potential allocation to a particular registrant (or TLD registry in the case of TLDs in the root). The name is not allocated, but could be if/when certain conditions are met.

3. Potential Variants

There are no script-wide variants in Cyrillic by nature of the Cyrillic code range: instead they arise at the level of language. As the root zone cannot use language-sensitive

[2] https://community.icann.org/download/attachments/16842778/Draft+Definitions.pdf

rules (e.g., reference to language tags), all labels in the script must share aggregate defined variant rules. Care should be taken in introducing variant Cyrillic characters in a TLD, as variant rules applicable to one language may not be applicable in another language.

When defining requirements, the needs of speakers of languages across the script must be taken into account. Note the potential for future collisions as new languages are recognized or spelling reforms occur: additions to a script may suggest a new variant rule. Policies need to allow for these scenarios, but cannot predict what form they will take: each state may assert the right to distinctive spellings, even as against the practice in closely related neighboring languages.

The choice of scripts, especially between Cyrillic and Latin, but also between Cyrillic and Perso-Arabic, remains politically fraught for some languages of the Russian Federation and states in the Central Asian region. Even where the choice of script remains stable, new spelling rules may be introduced. In practice, if there are changes, there will not be sufficient stability in the near term to establish variant relations between different spellings within a given language. Dozens of languages that use Cyrillic script have undergone spelling reforms in the last 100 years. Orthography for some of the languages is not yet stable suggesting that further reforms may occur.

Some examples of potential variant issues in Cyrillic are as follows.

3.1. <Ie> and <Io> in Russian

There is a special case in Russian language with characters <Ie> (“е”, U+0435) and <Io> (“ё”, U+0451).

In many (but not in all) words <Ie> is used as substitute character for <Io> - which is different from orthographically correct spelling in Russian language. The choice in such words whether to actually substitute <Io> with <Ie> could be (but is not always) determined by the context, and is orthographically wrong but acceptable for some words in cases of casual language usage. In other cases such substitution of the character will change the word’s meaning, and is unacceptable even as an exception (example «НЕБО»=sky, but “НЁБО»=palate).   <Io> is never a valid substitute for <Ie> in Russian. So, even if relationships between these two characters might look as variant-like, they are not symmetrical and these two letters are not considered equivalents in Russian language.

As an example, the Russian ccTLD registry operator does not consider these two code points variants, so they may be used as independent characters within the .рф TLD.

Besides, there are other languages that use the <Io> character (such as Belarusian language) and further research is needed whether the interchangeable use pattern identified above exists and is strong enough across all languages that use letter <Io> in their respective alphabets. In any case, <Io> as used in many languages of Asia (see Appendix A) is not in any variance relation with <Ie>.

3.2. <Ghe with upturn> and <Ghe> in Ukrainian

Ukrainian distinguishes between <Ghe>(“г”, U+0433) and <Ghe with upturn> (“ґ”, U+0491), which are accordingly 4th and 5th letters of the Ukrainian alphabet.

For other users of Cyrillic, these may be regarded as the same character, because the distinction is not made in those languages. Moreover, because letter “ґ” (U+0491) was eliminated from the Ukrainian alphabet in 1933 (unified with “г”) and reintroduced in 1990, even native speakers might not be able to accurately distinguish when “ґ” should be used.

It would generate substantial confusion if different operators were able to register otherwise similar TLDs with “г” (U+0433) and “ґ” (U+0491).

3.3. Cyrillic Small letter I with grave accent <045D>

The character “ѝ”, U+045D is shared between Bulgarian/Macedonian. Graphically, this is the CYRILLIC SMALL LETTER I WITH GRAVE. This is currently on newer keyboards but is not an official part of any alphabet. It is used phonetically to represent a stressed variant of the regular letter CYRILLIC SMALL LETTER I (“и” U+0438).
At the root level only, there is no need for this code point and it should probably not be allowed. If it is needed in the future, this character would need provisional rules – if used, it might be considered variant of ‘‘и” U+0438.

3.4 Old letters

Old letters no longer in use (example of CYRILLIC SMALL LETTER YAT U+0463) are particularly vulnerable to confusability (e.g. in this case CYRILLIC CAPITAL & SMALL LETTER SEMISOFT SIGN U+048C/U+048D). Such characters may be more troubling than they appear (as semi-obsolete signs), e.g. if used in trademarks.

3.5. Variance of a rarely used character with a sequence of commonly used characters

An example of this is U+04C7/04C8 (CYRILLIC CAPITAL & SMALL LETTER EN WITH HOOK), which is interchangeable in a specific language (Nenets) with U+043D+0433 (i.e. CYRILLIC SMALL LETTER EN followed by CYRILLIC SMALL LETTER GHE). Other differences with spelling may result in variants. For cases like this one, where an unusual character has a usual substitute, to avoid the possibility of variant issue one might disallow a particular character. It is worth noting that in some cases, a digraph may be confusable with a single character. This is included as a potential case.

3.6. Ukrainian Apostrophe <U+02BC>

Apostrophe in Ukrainian is a letter (sometimes referred as “quasi-letter”) with its use somewhat similar to Russian letter "ъ" (U+044A). Apostrophe is widely used in Ukrainian and cannot be omitted when writing.

The proper Unicode character for the Ukrainian apostrophe in Unicode is U+02BC (MODIFIER LETTER APOSTROPHE). The Unicode Script Property for this code point is “Common”.

U+02BC code point is part of the .УКР IDN ccTLD tables for second-level registrations. Due to complexity of entering U+02BC on the keyboard, it is common that people are using punctuation apostrophe (U+0027), which is not allowed in domain labels.

Besides, sometimes other code points are commonly used to indicate apostrophe, such as U+02BB (“MODIFIER LETTER TURNED COMMA”).

The use of U+02BC as part of a TLD label may be desired, but the corresponding security and stability implications need to be further evaluated.

Because new gTLD requirements say U-labels must have the same script property, with exception for certain orthographic rules, to be able to use U+02BC in TLDs it would be necessary to introduce an exception that would allow to make certain characters code points from outside the Cyrillic script block usable in Cyrillic labels.

3.7. Cyrillic letter З з - CYRILLIC CAPITAL & SMALL LETTER ZE

Cyrillic letter U+0417, U+0437 (З з - CYRILLIC CAPITAL & SMALL LETTER ZE) is visually similar to number 3 [DIGIT THREE U+0033]. Currently digits are not allowed in TLD labels, so at the moment the group does not envisage any variant-related issues for TLD labels. Should at some later point digits become allowed for the TLD labels, further security and stability issues may arise out of this visual similarity. The character does already occur in the Kazakhstan IDN ccTLD [қаз] approved by ICANN.

3.8. Composite characters

In addition, an area that would benefit from more research is that of Composite-character variants and Abstract Characters that currently do not have a code point assigned but are represented by multiple code points such as composite-characters in Kildin Sami or in Montenegrin Cyrillic script.

3.9. Within-script character visual similarity

Further research needs to be done to define the full list of visually confusable characters within the Cyrillic script. This will be discussed in the Integration and Solutions phases of the Variant Project.

There is a general consensus of the group that certain groups of code points may be confusable by users of different languages using Cyrillic scripts. (Good examples are the various forms of ghe, ka and en (e.g. U+0433, 0491, 0493; U+043A, 049B, 049D, 04A1; U+043D, 04A3, 04A5). There will be some which are confused by some users and not others. Therefore, a list of potential confusable code points would be beneficial. If a label contains one such character, other labels containing corresponding characters in the same position in the label might require special provisions.

Refer to Appendix A for examples of such characters. The large number of examples shows that this is a significant issue; however, the list of examples may well not be exhaustive.

4. Cross-script character visual similarity

There are several examples of Cyrillic and Latin and Greek character cross script visual similarity. Some of them are described in Unicode's mapping for visual confusables[3] for use in detecting possible security problems. However, the Unicode confusability algorithm does not seem to cover all cases of visual similarity. An example would be CYRILLIC CAPITAL LATTER TE (Т, U+0422) confusable with LATIN CAPITAL LETTER T (T, U+0054). Worse, CYRILLIC SMALL LETTER TE (т, U+0422) is also confusable with the Latin T, but when it is printed in italic is confusable with LATIN SMALL LETTER M (m, U+006D).

Another good source of information about visual similarity between Cyrillic, Latin and Greek characters is RFC 5992.

The following table provides some examples of cross-script similarity with Latin characters.

Latin	**Cyrillic**	**Case**	**Type**
Y (U+0059)	У (U+0423)	Upper	Similar
y (U+0079)	у (U+0443)	Lower	Identical
A (U+0041)	А (U+0410)	Lower and upper	Identical

[3] http://unicode.org/reports/tr39/#Confusable_Detection

Latin	Cyrillic	Case	Type
B (U+0042)	В (U+0412)	Upper	Identical
b (U+0062)	в (U+0432)	Lower	Not similar
M (U+004D)	М (U+041C)	Upper	Identical
m (U+006D)	м (U+043C)	Lower	Similar
3 (U+0033)	З (U+0417)	Upper	Similar
ć (U+0107)	U+0301, U+0441	Lower and Upper	Identical
m U+006D	*т* U+0442	Lower	Identical (only in italic)

The Cyrillic and Latin case study teams conducted a joint call on 8 September 2011, and reached general agreement that it is not clear to the teams that there is evidence of any valid circumstance under which it would be advised to allow mixed scripts between Cyrillic, Greek and Latin in a string when balanced against the significant issues that doing so introduces.

5. String-level issues

Cases of two labels that are different strings are considered out of scope for this team's report. As a general principle semantics are not considered as a basis for the identification of variants. The team believes that users will not expect labels in different languages to be variants. Lexical identity for different labels is not a basis for considering them to be in any type of variant relationship.

6. Types of Variant TLDs a Cyrillic script user would expect

6.1. Blocked variants

The Case Study team recommends blocking TLD labels that are visually confusable to any delegated labels, both intra and cross-script. This is the current ICANN policy with regards to TLD allocation and the group believes this approach must be preserved. Such labels should not be allocated at any time.

6.2. Reserved variants

The Case Study team recommends reserving variant TLD labels for all other variant cases in Cyrillic script as identified above. Such labels, if ever delegated in the DNS at all, can only be allocated to the registry that manages the corresponding fundamental label.

6.3. Other

Labels that are reserved pursuant to a Cyrillic variant-blocking policy may be delegated to the same registry operator provided all user experience implications arising out of variant Cyrillic characters are taken care of.

It should be noted that the vast majority of Cyrillic characters that are used in everyday life and business dealings are not in any kind of variant relationship. Variant issues described in this report manifest themselves in a very isolated class of situations.

From the business perspective the mere fact of inclusion of variant Cyrillic characters in a TLD label by an applicant seems to be a rather unlikely scenario.

Should such TLDs ever be delegated, and provided there is an ICANN policy in place that regulates parallel delegation (aliasing) of variant TLD labels, it may be a good idea to apply such a policy to Cyrillic TLD variant labels on the same basis as it would be applied to TLDs in other scripts.

However, the team believes that from both the technical and policy perspectives this is currently an imaginary, rather than a real life, scenario and therefore believes further elaborations on the applicability of aliasing and other similar techniques would not be appropriate for this specific case study team.

In this report we will therefore conclude that blocking or reservation are the primary methods of dealing with variant characters in Cyrillic script.

7. Evaluation of TLD Applications with variants

The case study team believes that ICANN should take a conservative approach in evaluating TLD applications that contain Cyrillic characters in the TLD label. The team recommends that ICANN take an inclusion-only approach and only accept Cyrillic characters that have been vetted by the respective language communities.

7.1. The need for Variant Tables for the root

To standardize all IDN TLD implementations, Variant Tables (or a similar tool) for the root are needed. The initial version of this table needs to be restrictive until there is input that gives a basis for understanding the issues related to the addition of new code points.

7.2. Whether IDN variants at TLD level should be based on language or script

As Top Level Domains are shared between users of multiple languages, and because language cannot be an attribute of a DNS label, we believe Cyrillic TLD labels should be script-based, rather than language-based.

In the Cyrillic space multiple language communities share a number of Cyrillic characters. As we showed above, many languages use some of the Cyrillic characters in their unique way which may introduce non-obvious variant-related issues.

It is therefore important that Cyrillic TLDs are at time of evaluation vetted by experts representing all language groups that use specific characters included in the TLD label.

7.3. Considerations for a process to define the root Variant Tables

Root Variant Tables must be defined after a consultation with the community and the relevant entities responsible for all the languages using the Cyrillic script. If there is to be a root variant table, there should be a process to develop such a table. We do not know what that process should be.

8. Impact of Variants on Registry/Registrar Operations

The case study team recommends reservation or blocking as preferred ways of dealing with Cyrillic variant characters in TLD labels. As a consequence, the team does not envisage any impact on registry/registrar operations.

9. Other Considerations

This section describes issues that are not directly related to variants, but the team believes they are important for consideration of IDNs at the root level.

9.1. Code point series used interchangeably, but not captured by Normalization Form C.

There are few abstract characters in Cyrillic scripts that are composed of multiple code points. For example, in Montenegrin Cyrillic script, there are two newly added characters that are composed of multiple code points pairs: U+0437 U+0301 (CYRILLIC SMALL LETTER ZE plus COMBINING ACUTE ACCENT); and U+0441 U+0301 (CYRILLIC SMALL LETTER ES plus COMBINING ACUTE ACCENT); These two characters have only recently been added to the Montenegrin Cyrillic script by a revision of orthography rules published by the Ministry of Education and Science in 2009, (http://www.gov.me/files/1248442673.pdf). Since they do not exist as a single code point in Unicode, both pairs can be used together to compose a new character according to IDNA2008 and part of that requirement is that the pairs must be stable under Unicode Normalization Form C (NFC). Normalization is transcoding one set of code points to another, and NFC is required by IDNA2008.

However, it should be noted that the second part of the multiple code point pair, used together with the base character, the code point U+0301(combining acute accent), does not belong to the Cyrillic block, but instead has the Script Property Inherited.

In addition to that, the new character formed in Cyrillic U+0441 U+0301 appears identical to a character in Latin: LATIN SMALL LETTER C WITH ACUTE, (ć, U+0107). Montenegrin uses both Cyrillic and Latin alphabets, so it could be added to the cross scripts confusing table. (See page 8 - http://www.gov.me/files/1248442673.pdf), character number 23 in the Cyrillic table and character number 5 in the Latin table.)

The new orthography rule confirms that some Montenegrin words starting with composite character U+0441 U+0301 have previously been presented by two separate and non-composite characters U+0441 and U+0458 and only a native speaker would know when the two separate characters are pronounced as separate characters and when as a single character, so it poses an interesting example where characters are not variants all the time.

So, a newly added multiple code character U+0441 U+0301 с́ until recently used to be represented in writing by two characters U+0441 and U+0458 “с” and “ј”, so the words ćутра and сјутра (meaning tomorrow) could now be written either way, but only the native speaker would know the difference: the pronunciation is slightly different. However they have the same meaning and are therefore semantic variants and they should not necessarily be treated as Character variant Labels, as they are not consistent throughout.

9.2. Inclusion of Characters outside the Cyrillic script block

A number of code points that are not part of the Cyrillic script block may be considered to be available for Cyrillic top-level labels. The team recommends that explicit exceptions to the prohibition on mixing scripts in top-level labels should be made to allow specified code points with the Script Property of Common or Inherited that are part of the established orthography for languages that use Cyrillic, for example:

- U+02BC
- U+0301

However, the team recommends additional study on the security and stability implications before actual inclusion, and expects that this will be discussed during the Integration and Solutions phases of the Variant Project.

10. Conclusions

Based on the work of this team it appears that there are possible cases of variant characters in languages using the Cyrillic script.

The group agreed that the most preferable solution is to be conservative, as due to the nature of the DNS, it is easier to open up additional options in the future, than to introduce liberal rules in the beginning and subsequently make them more restrictive.

The group also identified that there might be a good reason to introduce exceptions to make certain code points from outside the Cyrillic script block usable in Cyrillic labels, i.e., by treating them as part of the script block and making them possibly available for use in root zone labels including, for example, U+02BC and U+0301.

Another outcome of the discussion is consensus that blocking is the preferable way to implement variants.

Due to the large number of languages that constitute the Cyrillic script, special care should be taken when introducing variants into the root as they affect all languages in the script, and to avoid unintended consequences. None of the issues identified in this report should prevent the future delegation of Cyrillic script TLDs.

11. Case Study Team

The Cyrillic case study team initially met face to face at the ICANN meeting in Singapore in June 2011, and conducted weekly one hour calls each week through the month of September 2011. The Cyrillic team also conducted a face-to-face meeting hosted by UNESCO in Paris on 20-21 September to finalize the report for the Cyrillic case study. Additional calls were conducted on 29 September and 6 October 2011. The report represents the best efforts of the case study team to identify the issues raised by the Cyrillic script.

Team members:

- Alexei Sozonov (Coordinator)
- Alexey Mykhaylov
- Oleksiy Ptashniy
- Daniel Kalchev
- Iliya Bazlyankov
- Oksana Prykhodko
- Saso Dimitrijoski
- Sergey Sharikov
- Vladimir Shadrunov
- Desiree Miloshevic (Observer)
- Yuriy Kargapolov (Observer)
- Irmgarda Kasinskaite-Buddeberg (Observer on behalf of UNESCO)

The team thanks external experts and ICANN staff for their invaluable professional advice and for their hard work during the team meetings. We would like to extend our thanks and gratitude to:

- Andrew Sullivan, DNS expert
- Nicholas Ostler, linguistic expert
- Francisco Arias, ICANN staff
- Patrick Jones, ICANN staff
- Karen Lentz, ICANN staff
- Naela Sarras, ICANN staff

Appendix A: Overview of Potential IDN Variants in Cyrillic-derived Scripts

[Nicholas Ostler <nicholas@ostler.net> version 3: 21 September 2011]

The list for each language is as extensive as we could make it, but we cannot confirm that this represents an exhaustive list of potential variants. For such an exhaustive list, experts on the individual languages must be consulted (both on their own languages, and on the pairs of glyphs that might appear to them as potentially confused variants in other languages).

Language	**No. of letters**	**Distinctive letters**	**Possible Variants**
European Slavonic			
- Church Slavonic	44	Ѕѕ Ҫҫ Ѹѹ Ѡѡ Щщ Ѣѣ □ꙗ Ѥѥ Ѧѧ Ѫѫ Ѩѩ Ѭѭ Ѯѯ Ѱѱ Ѳѳ Ѵѵ	Fita (Ѳ, ѳ) U+0472,U+0473 ≠ barred o (Ө, ө) U+04E8,U+04E9 in Mongolian & Turkic lgs
- Bela-rusian	33	Ёё Ii Ўў ’	Some users often substitute Ee for Ёё. ’ U+02BC ≠ ’ U+0027 (punctuation apostrophe); Ўў ≠ Tajik Ӯӯ.
- Bosnian	30	Ђђ, Јј, Љљ, Њњ, Ћћ, Џџ	
- Bulgarian	31	Щщ ѝ	U+045D (Example: To her mother. / На майка ѝ.)
- Mace-donian	31	Ѓѓ Ѕѕ Јј Љљ Њњ Ќќ Џџ ѝ	dzhe Џџ U+040F,U+045F ≠ tse Цц U+0426,U+0446, U+045D
- Monte-negrin	32	Ђђ З́з́ Јј Љљ Њњ С́с́ Ћћ Џџ	Зз ≠ З́з́, Сс ≠ С́с́: the latter have an acute accent, in upper and lower case. Their Unicode is not yet set as pre-composed characters.
- Russian	33	Ёё Щщ Ээ Юю Яя	Some users often substitute Ee for Ёё.
- Rusyn	37	Ґґ Єє Іі Її (Ѣ obsolete)	Ґґ U+0490,U+0491 ≠ Гг Іі U+0406,U+0456 ≠ Її U+0407,U+0457 ≠ ‘palochka’ Ӏӏ U+04C0,U+04CF
- Serbian	30	Ђђ, Јј, Љљ, Њњ, Ћћ, Џџ	
- Ukrainian	34	Ґґ Єє Іі Її ’ (Ѣ obsolete)	Ґґ U+0490,U+0491 ≠ Гг Іі U+0406,U+0456 ≠ Її U+0407,U+0457 ≠ ‘palochka’ Ӏӏ U+04C0,U+04CF’; U+02BC ≠ ‘ U+0027 (punctuation apostrophe)
Romance			

Moldovan (in Transnistria)	34	Ӂӂ	U+04C1,U+04C2. But can be composed of U+0416,U+0436 + U+0306 breve.
Caucasian			
-All (except Abkhaz), viz 10: - Avar - Chechen - Ingush - Dargwa - Lak - Lezgian - Tabasaran - Abaza - Adyghe - Kabardian	34	‘palochka’ӀӀ U+04C0,U+04CF marking glottalization, Ёё,	палочка = ‘stick’ ӀӀ U+04C0,U+04CF is often replaced with a capital Latin letter I, small Latin letter l or the digit 1. Furthermore, upper and lower case palochka are not visually distinct.
- Abkhaz	40 (offic. 58, but 18 are di-graphs)	Ҕҕ, Ц̡ц̡, Ҽҽ, Ҿҿ, ә, Ӡӡ, Қ қ, Ҟҟ, Ԛԛ, Ҧҧ, Ҭҭ, Ҳҳ, Ҵҵ, Ҷҷ	They don’t use palochka, hence the profusion of different forms for glottalized consonants! Most are potential look-alikes: to Бб, Цц, Єє, Єє, Latin schwa, Зз, Кк, Кк, Latin Q, Latin M, Тт, Хх, Цц, Чч
Iranian			
- Ossetic	25	Ææ, Ёё	
- Tajik , & - Yaghnobi	35	Ғ ғ, Ёё, Ӣӣ, Ққ, Ӯӯ (U+04EE,04EF), Ҳҳ, Ҷҷ	≠ Гг, Йй, Кк, Belarusian/Dungan Ўў, Хх, Чч.
Indic			
- Romani (Kalderash)	32	Ғғ	≠ Гг ≠ Ґґ
- Romani (Ruska Roma)	32	Ґґ	≠ Гг ≠ Ғғ
Chinese			
- Dungan	36	Җҗ, Ёё, Ңң, Әә, Ўў, Үү	Җҗ ≠ Жж, Ңң ≠ Нн, Ўў ≠ Tajik Ӯӯ (U+04EE,04EF), ≠ Altay Ӱӱ (U+0423,0443), Үү ≠ Уу.
Mongolian			
- Buryat	32	Ёё, Өө, Үү, Һһ	Өө (U+04E8,U+04E9) ≠ Old Church Slavonic fita Ѳ); Үү ≠ Уу, Һһ ≠ Latin h

- Kalmyk	34	Җҗ, Ңң, Өө, Үү, Һһ	as above (Dungan and Buryat)
- Mongolian (Khalkha)	35	Ёё, Өө, Үү	
Turkic			
- Altay	37	Jj, Ҥҥ, Ӧӧ, Ӱӱ	Jj ≠ Latin Jj; Ҥҥ (U+04A4,04A5) unique to Altay, Mari and Sakha (and Aleut); Ӧӧ, Ӱӱ (U+041E,043E; U+0423,0443) can alternatively be composed with U+0308. Ӱӱ ≠ Chuvash Ӳӳ (U+04F2,04F3) ≠ Ўў (U+040E,045E).
Balkar = Karachay-Balkar			
- Bashkir	41	Ғғ, Ёё, Ҙҙ,Ҡҡ, Ңң, Өө, Ҫҫ, Үү, Һһ, Әә	Ҙҙ, Ҫҫ cannot be composed in Cyrillic, but evidently Ҫҫ looks identical to Latin Çç (C-cedilla)
- Chuvash	37	Ӑӑ, Ёё, Ӗӗ, Ҫҫ, Ӳӳ	Forms with breve can be composed. Also Уу (U+0423, 0443) with double acute U+030B. All have homographs in Latin.
- Kazakh	42	Әә, Ғғ, Ёё, Ққ, Ңң, Өө, Ҫҫ, Ұұ,Үү, Һһ, İi	As above, e.g. Dungan, Buryat, Ukrainian. Ұұ (U+04B0, 04B1) is apparently unique to Kazakh.
- Karachay-Balkar	34	Ўў (U+040E,045E).	≠ Ӱӱ (U+0423,0443) ≠ Chuvash Ӳӳ (U+04F2,04F3)
- Khakas	39	Ғғ, Ңң, Ӧӧ, Ӱӱ, Ӌӌ, Әә	≠ Гг, Нн; Ӧӧ, Ӱӱ (U+041E,043E; U+0423,0443) can alternatively be composed with U+0308. Ӱӱ ≠ Chuvash Ӳӳ (U+04F2,04F3) ≠ Ўў (U+040E,045E). Ӌӌ (U+04CB,04CC) ≠ Чч.
- Kumyk	33	-	No distinct letters: but Kumyk posits a number of digraphs: Гъ гъ, Гь гь, Къ къ, Нг нг, Оь оь, Уь уь. It is not alone in this; but since such digraphs do not have separate codes, or even conjoined rendering, they may be irrelevant to Unicode.
- Kyrgyz	30	Ёё, Ңң, Үү	Ңң ≠ Нн, Үү ≠ Уу.
- Nogai	31	-	No distinct letters: but Nogai posits a number of digraphs: Аь аь, Нъ нъ, Оь оь, Уь уь. It is not alone in this; but since such digraphs do not have separate codes, or even conjoined rendering, they may be irrelevant to Unicode.

- Tatar	39	Әә, Ж̱ж, Ңң, Өө, Үү, Һh	Ж̱ж ≠ Жж, Ң̱ң ≠ Нн, Ү̱ү ≠ Уу, Һh ≠ Latin h, Әә ≠ Latin schwa
- Tuvan		ң, Өө,Үү	ң ≠ н, Өө ≠ O.C.Sl. Fita (Ѳ, ѳ),Үү ≠Уу
- Sakha (aka Yakut)	38	Ҕҕ, Ҥҥ, Өө, Һһ, Үү	Ҥҥ (U+04A4,04A5) - unique to Altay, Mari and Sakha (and Aleut).
- Uzbek	30	Ёё, Ўў, Ққ, Ғғ, Ҳҳ	Cf Tajik, though with Ўў not Ӯӯ.
Uralian			
- Nenets	35	Ӈӈ(U+04C7, 04C8),	э has two pronunciations, as [e] and [æ]. www.omniglot.com/writing/nenets.htm suggests that the latter can be marked with a superposed dot. Unicode appears to know nothing of this. According to the same source, Ӈӈ(U+04C7, 04C8) has a variant as a digraph: НГ нг
- Khanty	53	Әә, Ӛӛ, Ққ, Ӆӆ, Ԓԓ, Ӊӊ, Ӈӈ, Ӧӧ, Өө, Ӫӫ Ӱӱ, Ўў, Ҳҳ, Ӌӌ, Є̱є, Є̈є̈ (U+04EC, U+04ED), Ю̆ю̆, Я̆я̆ (i.e. LETTERS YU and YA with BREVE U+0306)	Separate composition is possible for latters with diacritic umlaut and breve.
- Komi-Zyrian	34	Ёё, Ӧӧ	Separate composition is possible. Komi-Yazvin alphabet was proposed in 2003, including also I̱i, Өө, Ӱӱ
- Mari	37	Ӓӓ, Ёё, Ӧӧ, Ӱӱ, Ӹӹ, Ҥҥ	Separate composition is possible for letters with diacritic umlaut. Ҥҥ (U+04A4, 04A5) - unique to Altay, Mari and Sakha (and Aleut).
- Erzya (aka Mordvin)	33	-	
- Moksha	33	-	
- Kildin Sami	45	Ӓӓ, ', (Һh), Ҋҋ, (Jj), Ӆӆ, Ӎӎ, Ӊӊ,	Use of apostrophe. Separate composition is possible for letters with diacritic umlaut,

		Ӊӊ, Ҏҏ, Ҍҍ, Ӭӭ	not for those with tails or cedillas. . Evidently the tailed letters are hard to distinguish in small fonts from plain equivalents л, м, н, р, э. In some styles, ' is replaced by Һh, and Ҋҋ by Jj.
- Udmurt	38	Ӝӝ, Ӟӟ, Ӥӥ, Ӧӧ, Ӵӵ	Various letters, with umaluts, can be separately composed.
Siberian			
- Chukchi	36	Ӄӄ, Ԓԓ, Ӈӈ, '	Use of apostrophe.
- Koryak	37	В'в', Ӄӄ, Ӈӈ	Use of apostrophe.
- Itelmen	40	Ӑӑ, К’к’, Ӄӄ, Ӄ’ӄ’, Љљ, Ӆӆ, Њњ, Ӈӈ, О̆о̆, П’п’, Т’т’, Ч’ч’	Use of apostrophe. Separate composition is possible for letter with breve.
- Even	38	Ӑӑ, Ӈӈ, Өө, Ӫӫ, Ӧӧ	Separate composиtion is possible for letters with umlaut. Өө (U+04E8,U+04E9) ≠ Old Church Slavonic fita Ѳ);
- Evenk	34	Ӈӈ	
- Nanai	33	-	
- Ket		Ӷӷ, Ӄӄ, Ӈӈ, Өө, ’	Өө (U+04E8,U+04E9) ≠ Old Church Slavonic fita Ѳ); Use of apostrophe.
- Nivkh	46	Ӷӷ, Ғғ, Ӻӻ, К’к’, Ӄӄ, Ӄ’ӄ’, Ӈӈ, П’п’, Ҏр̌ , Т’т’ , Ӽӽ, Ӿӿ, Ч’ч’	Use of apostrophe.
- Yukaghir (Tundra)	35	Ғғ, Ӧӧ	Separate composition is possible for letters with umlaut.
- Yukaghir (Kolyma)	39	Җҗ, Ҕҕ, Ӄӄ, Ӈӈ, Өө, Әә	Evidently, Өө and Әә are similar.
Eskimo Aleutian			
- Aleut	39	г̂, ҟ, ӈ, ў, х̂, ө, ѵ	ҟ shared with Abkhaz. Those with inverted breves not evidently supported as separate characters. Өө (U+04E8,U+04E9) ≠ Old Church Slavonic fita Ѳ). Apparently the only language still

			to use O. Ch. Sl izhitsa (ѵ) which looks like Latin v.
- Alutiiq (aka Sugpiak)			Not evidently written in Cyrillic.
- Yuit (aka Central Siberian Yupik)			Not evidently written at all.
Restricted to Alaska			
- Tlingit			“the population familiar with both the Cyrillic script and the Tlingit language is rather small, thus no such script is likely to find serious use.” Wikipedia

Appendix B

The case study team spent some time discussing the concept of Alternate Names and decided to move this to Appendix B for future consideration. This section is included in the report for information, but the case study team has decided not to recommend use of alternate names in Cyrillic.

Alternate Names

When it is desirable to make alternate names active at the same time, there are two techniques now available in the DNS. The first is to use provisioning to ensure that the different names resolve the same way. In the absence of sophisticated provisioning software, it will not be possible to guarantee the equivalence of the DNS trees; but as long as the provisioning software provides this support, the different DNS trees can be kept in sync with one another (subject to the usual conditions of loose coherence of the DNS).

The alternative technique is to use DNAMEs for all the alternates but one. Under this mechanism, one of the names becomes the bottom of the canonical tree, and all the other names are aliases of that name. Putting a DNAME in the root zone has not been tried before, and there may be issues in employing such an innovation; but there is no apparent protocol reason why this would not work.

In either case, while these techniques may make the names work effectively as alternates of one another from the DNS point of view, they are not magic. Many services need to know their own names. Each such service that is expected to operate at any of the alternates will need to be configured to accept traffic using that name; these services include things like HTTP, SMTP, and SSL/TLS support for nearly every protocol. There is no support in the DNS for linking names together bi-directionally (regardless of the alternate technique one uses), so this configuration step necessarily requires out of band communication and knowledge of how names are supposed to interoperate. Moreover, in the aliasing case, some of the names cannot be used as the target of an MX (and other such uses). This makes the use of alternates more difficult than they might seem at first glance, and suggests that aliasing in particular is not as usable as one would like. System administrators are users of variants just as much as those accessing web sites, and the effects of these techniques on the network environment needs to be considered before activating any variant name.

This issue is nothing new: almost no servers configure themselves automatically on start up by querying the DNS, and there is reason to believe that such an approach is at least as dangerous as it is helpful. But the complication of an environment where many servers are known by several names (as would happen under the most generous plans for variants) might make the problem more pressing than it has been in the past. In effect, a variant-rich environment will require the most basic system administrator to become familiar with tools used today mostly by ISPs and hosting service providers.

Another problem is abuse prevention. The purpose of the alternate names is in many cases to reduce user confusion and lead user to the same website if one of the alternate names is used. However it is likely that technically (either by analyzing the domain user used or referral http referrer on the webserver) it will be possible to setup different individual websites for each of the alternate names. It might be possible to reduce abuse by developing appropriate technical solutions or via tight cooperation with the browsers; however, there is no clear enforcing mechanism at the moment.