IDN Variant TLD Implementation: Appendices

Contents

Background

The current document is part of the set of the following six documents released for public comment:

A. IDN Variant TLD Implementation – Executive Summary
B. IDN Variant TLD Implementation – Motivation, Premises and Framework
C. IDN Variant TLD Implementation – Recommendations and Analysis
D. IDN Variant TLD Implementation – Rationale for RZ-LGR
E. IDN Variant TLD Implementation – Risks and their Mitigation
F. IDN Variant TLD Implementation – Appendices (A: Definitions, B: Use of ROID, C: Limiting Allocated Variant TLDs)

Appendix A. Glossary

Throughout this document, the following writing conventions are used. Much of the terminology depends on the definitions in IIR, and unfamiliar terms not listed here should be found in the IIR Glossary.

Labels and names

A top level label = t1
Top level Variant 1 = t1v1
Top level Variant 2 = t1v2
...
A second level label = s1
Second level Variant 1 = s1v1
Second level Variant 2 = s1v2

Other labels down the tree (third and ...)
Another level label = a1
Another level variant label = a1v1

In general, this document does not distinguish the fundamental label from its variants. Instead, the fundamental label is the one that is under consideration at any time. When two or more labels are variants of each other, then they are identified as tv1, tv2, tv3 or sv1, sv2, sv3, and so on. Moreover, nothing about this document should be understood as expressing the view that there is anything technically special about root zone or second level allocations. These are merely the parts of the DNS where ICANN can exert some policy influence, so they are distinguished for that reason.

Note that under the recommendations in this document, these are all examples of possible cases that would be covered by the policy:

- t1v1, t1v2, t1v3, t1v4 …
- sv1.t1v1, sv1.t1v2, sv1.t1v3 …

These are *not* examples of a case that would be covered by the policy:

- sv1.t1v1, sv2.t1v1, sv3.t1v1, ...

Label States

Label states were defined in IIR. Their essence is copied here.

Blocked: A status of some label with respect to a zone, according to which the label is unavailable for allocation to anyone. The term “to block” denotes the registry (the zone operator) taking this action.

Withheld: A status of some label with respect to a zone, whereby the label is set aside for possible allocation to some entity. In this strict sense, a withheld name is not actually allocated. The term “to withhold” denotes the registry (the zone operator) performing the setting aside.

Allocated: A status of some label with respect to a zone, whereby the label is associated administratively to some entity that has requested the label. This term (and its cognates “allocation” and “to allocate”) represents the first step on the way to delegation in the DNS. When the registry (zone operator) allocates the label, it is effectively making a label a candidate for activation. Allocation does not, however, affect the DNS at all.

Activated/Active: A status of some label with respect to a zone, indicating that there are DNS resource records at that node name; or else that there are subordinate names to that name, even though there are no resource records at that node name. In the case where there are resource records at the node name, any resource record will do. In the case where there are subordinate names but no resource records (except those to support DNSSEC), the label names an empty non-terminal. A registry (zone operator) setting the active status activates the name, or performs activation.

Delegated: A status of some label with respect to a zone, indicating that in that zone there are NS resource records at the label. The NS resource records create a zone cut, and require an SOA record for the same owner name and corresponding NS resource records in the subordinate zone. The act of entering the NS records in the zone at the parent side of the zone cut is delegation, and to do that is to delegate. This definition is largely based on RFC 1034; the reader should consult RFC 1034 for detailed discussion of how the DNS is broken into zones.

An additional state is defined:
Withheld-same-entity: A Withheld label is set aside for possible allocation to only the same entity of the labels in the variant set. See Section 6.

Other definitions

IDL set: A label whose code points are all included in the zone repertoire, along with all of the labels arising from the application of the code point variant rules on that first label. (From IIR.)

RDDS: Registration Data Directory Services (e.g., WHOIS, web Whois, RDAP)

ROID: see section 2.8 of RFC 5730

Registry Operator: Term used by ICANN to name the party signing a registry agreement for a gTLD.

Appendix B. Analysis of Implementing the Repository Object Identifier (ROID) to Satisfy the "Same Entity" Requirement

Introduction

As explained in Section 2.3 of SAC-060, variant names that ICANN org delegates[1] must have the "same entity" as the name holder (usually called "Registry Operators" at the TLD level, and "registrants" at any other level) in order to minimize security risks.

This report examines the use and reliability of implementing the Repository Object Identifier (ROID) to satisfy the "same entity" requirement in gTLDs. To meet this expectation at the top level, it is recommended that any variant of a gTLD or a ccTLD label is allocated to the same Registry Operator as the primary TLD label.

At the second level and below, it is also recommended ensuring the label *s1* and its variant labels under all variant TLDs (e.g., *s1.t1*, *s1.t1v1, s1v1.t1*, *s1v1.t1v1,* etc.) are allocated to the same entity. This should be achieved by having the same ROID for the registrant of each variant label registered for the gTLDs. However, most ccTLDs do not have a contractual agreement with ICANN org and may not implement ROIDs. In such cases, it is recommended that ccTLDs identify the "same entity" by some other mechanism, coordinated by the ccNSO.

Background on ROIDs

A ROID is a globally unique identifier that is generated by the repository. A registry can have multiple TLDs in the same repository, so a repository can encompass one or more TLDs based on the choice of the registry. Thus, it is guaranteed to refer to the same contact object in the registry.

Per Specification 6 of the new gTLD registry agreement and the Functional Specifications Appendix of most legacy gTLDs, according to RFC 5730 (http://tools.ietf.org/html/rfc5730), a globally unique identifier must be assigned by the registry to every object when the object is created. Registries must also register their Extensible Provisioning Protocol (EPP) Repository identifier with Internet Assigned Numbers Authority (IANA) (https://www.iana.org/assignments/epp-repository-ids). The globally unique identifier is a concatenation of the local identifier for a

[1] Normally this is "from the root zone," but the principle is stated this way to accommodate any case where ICANN delegates from some other zone.

contact object, followed by a hyphen ("-"), followed by the repository identifier, e.g. 5372808-EXAMPLE (see the example at the end of this appendix).

Per the Registry Agreement, the use of ROIDs is required for some objects, at least, in the RDDS output, data escrow, BRDA, EPP and Trademark Database (TMDB) List of Registered Domain Names (LORDN) files.[2] In the WHOIS output, the fields "Registry Admin/Tech/Billing/Registrant ID:" refer to the ROID for the contact object associated with a domain. These fields are highlighted in yellow in the example output for the query objects shown at the end of this appendix.

These RDDS fields are part of the minimum output requirements for the query objects which are published and made publicly available on RDDS. Registries and registrars are currently required to display this data in the public WHOIS.[3] In order to verify and ensure the operational stability of Registry Services as well as to facilitate compliance checks on accredited registrars, Registry Operators provide ICANN org on a weekly basis with up-to-date thin registration data, referred as BRDA by the new gTLD registry agreement. This data does not include the contact information related to the registrant, which falls under the scope of protection of the GDPR.

Although the requirement is for "thin" data, most gTLDs currently provide "thick" or "full" data. Looking at the IDN gTLDs that provide full BRDA to ICANN org, which is 65% of the 94 IDN gTLDs, all of them follow the ROID format in their contact ROID fields. However, 29% of those that provide full BRDA have at least one contact with a ROID suffix that is not registered with IANA. A communication with the affected TLDs was initiated, the registries acknowledged the issues, and are expected to fix them.

ROIDs are stored in the Shared Registry System (SRS), a system for managing a shared domain name registry which allows multiple registrars to make changes to a registry simultaneously. Furthermore, the SRS provides the functions required to support all of the usual business functions of a domain registration service including the creation, maintenance, querying and deletion of domain details, querying of public details of a domain to support the RDDS, and transfer of domains between registrars. Registrars generate most of the work that the SRS server handles. Generally, the registry will be responsible for maintaining all of the information associated with domain name registrations (including both the technical information required to produce zone files and the contact information for registrars and registrants), running the SRS service to support registrars' and its own maintenance of domains, and running the RDDS.

---

[2] ICANN, “gTLD Registry Advisory: Correction of non-compliant ROIDs,” (2015), accessed 1 January 2018, https://www.icann.org/resources/pages/correction-non-compliant-roids-2015-08-26-en

[3] ICANN, “Registry Registration Data Directory Services Consistent Labeling and Display Policy,” (2017), accessed 1 January 2018, https://www.icann.org/resources/pages/rdds-labeling-policy-2017-02-01-en

Impact of Implementing IDN Variants on the ROID Requirement

At the top level, having the same entity for two variant TLDs can be achieved by ensuring that the Registry Operator is the same. In practical terms, this is achieved by ensuring the Registry Operator (including the name and address) for the two variant TLDs is the same, and that is reflected in the root zone RDDS operated by Public Technical Identifiers (PTI) – an affiliate of ICANN org.

At the second level and below, ensuring the same label *s1* and its variant labels under all gTLD variants (e.g., *s1.t1*, *s1.t1v1, s1v.t1*, *s1v1.t1v1,* etc.) are allocated to the same entity could be achieved by having the same ROID for the registrant of all such domain names. Since the ROID is a globally unique identifier that is generated by the repository with multiple TLDs in the same repository, it can be set up to refer to the same contact object in the registry.

Per the Registry Agreement, the use of a unique-per-object ROID is required, however the same contact ROID may not be assigned for the same registrant across gTLDs. To ensure that registries reuse contact objects for the same registration, each Registry Agreement (including the existing contract) must contain provisions requiring ROIDs for the same registrant across gTLD variant sets.

This requirement is to be followed by Registry Operators and can be verified by looking at the RDDS output and comparing one field (which implies the rest of the registrant fields are the same). This requirement also has an impact on the registrars, which manage these registrations, thus registries would also need to amend its agreement with each registrar and work with registrars to determine ways for maintaining the same entity for all variant labels under all variant TLDs during registration, transfer, dispute resolution and other relevant processes.

In order to verify that the same second-level label beneath all variant labels is allocated to the same entity, compliance checks could be performed as needed given that registrant ROIDs and other registrant fields are shown in RDDS. While privacy regulations in the European Union (EU) will likely affect how WHOIS data is published, including the ROID field which may no longer be displayed in public WHOIS, it would not impact ability to conduct regular checks. Monitoring could still take place if ICANN org is accredited for access to full WHOIS data for this specific purpose as long as it limited to what is necessary. In addition, pro-active monitoring could take place by, for example, reviewing a random sample at defined intervals by using a combination of BRDA and RDDS queries. Lastly, compliance could also check for the “same entity” requirement during regular compliance audits.

Some registrars in practice may not reuse contact objects for different registrations, thus registrars would need to support the requirement. Furthermore, some registries may not want to use ROIDs and may define their own mechanism for satisfying the “same entity” requirement. This may include the IDN ccTLDs. In regard to gTLDs, the mechanism proposed should be evaluated at the time of application for technical feasibility and compliance potential, on a case

by case basis. In regard to ccTLDs, ICANN org should work with ccNSO to request its members to develop a common definition of “same entity” based on a well-defined subset of the registration data and encourage its members to adopt it consistently in case they implement IDN variant TLDs.

It is recommended that at the second level and below, ccTLD operators should ensure the same label *s1* and its variant labels under all ccTLD variants are allocated to the same registrant. While the use of ROIDs to satisfy the “same entity” requirement is based on contractual requirements in the agreements between ICANN org and registrars and registries, ccTLD policies regarding registration, registrars and RDDS are managed according to the relevant governance mechanisms within the country. Given most ccTLDs do not have a contractual agreement with ICANN org and are not required to use ROIDs, the ICANN org does not have contract authority to perform compliance checks against ccTLD operators.

Conclusion

In sum, it is recommended that any variant of a gTLD or a ccTLD label is allocated to the same Registry Operator as the primary TLD label. At the second level and below it is also recommended ensuring the label *s1* and its variant labels under all variant TLDs (*s1.t1*, *s1.t1v1, s1v1.t1* and *s1v1.t1v1*) are allocated to the same entity. This should be achieved by having the same ROID for the registrant of each variant label registered for the gTLDs, or an alternate mechanism, approved on a case by case basis for gTLDs not employing ROIDs consistently and through a common mechanism agreed within ccNSO for the IDN ccTLDs.

In the WHOIS system, compliance checks could be performed as needed given that registrant ROIDs and other registrant fields are shown in RDDS. While the privacy regulations in the EU will likely affect how registration data is displayed, monitoring could still take place through the following channels: (a) ICANN org could be accredited to access additional registration data for this specific purpose as long as it limited to what is necessary; (b) ICANN org could access the contact information in its Registration Data Escrow deposits; and (c) Compliance could also check for the “same entity” requirement during regular compliance audits.

Domain Name Data[4]

- **Query format:** whois EXAMPLE.TLD

- **Response format:**
  Domain Name: EXAMPLE.TLD
  Registry Domain ID: D1234567-EXAMPLE
  Registrar WHOIS Server: whois.example-registrar.tld
  Registrar URL: http://www.example-registrar.tld
  Updated Date: 2009-05-29T20:13:00Z
  Creation Date: 2000-10-08T00:45:00Z
  Registry Expiry Date: 2010-10-08T00:44:59Z
  Registrar Registration Expiration Date: 2010-10-08T00:44:59Z
  Registrar: EXAMPLE REGISTRAR LLC
  Registrar IANA ID: 5555555
  Registrar Abuse Contact Email: email@registrar.tld
  Registrar Abuse Contact Phone: +1.1235551234
  Reseller: EXAMPLE RESELLER1
  Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
  Domain Status: clientRenewProhibited https://icann.org/epp#clientRenewProhibited
  Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
  Registry Registrant ID: 5372808-EXAMPLE[5]
  Registrant Name: EXAMPLE REGISTRANT
  Registrant Organization: EXAMPLE ORGANIZATION
  Registrant Street: 123 EXAMPLE STREET
  Registrant City: ANYTOWN
  Registrant State/Province: AP
  Registrant Postal Code: A1A1A16
  Registrant Country: AA
  Registrant Phone: +1.5555551212
  Registrant Phone Ext: 12347
  Registrant Fax: +1.5555551213
  Registrant Fax Ext: 4321
  Registrant Email: EMAIL@EXAMPLE.TLD
  Registry Admin ID: 5372809-EXAMPLE[6]
  Admin Name: EXAMPLE REGISTRANT ADMINISTRATIVE
  Admin Organization: EXAMPLE REGISTRANT ORGANIZATION
  Admin Street: 123 EXAMPLE STREET
  Admin City: ANYTOWN
  Admin State/Province: AP
  Admin Postal Code: A1A1A1
  Admin Country: AA
  Admin Phone: +1.5555551212
  Admin Phone Ext: 1234
  Admin Fax: +1.5555551213
  Admin Fax Ext: 1234
  Admin Email: EMAIL@EXAMPLE.TLD

[4] ICANN, “Registry Registration Data Directory Services Consistent Labelling and Display Policy.”
[5] Note: “EXAMPLE” represents one or more TLDs.
[6] Note: “EXAMPLE” represents one or more TLDs.

Registry Tech ID: 5372811-EXAMPLE[7]
Tech Name: EXAMPLE REGISTRANT TECHNICAL
Tech Organization: EXAMPLE REGISTRANT LLC
Tech Street: 123 EXAMPLE STREET
Tech City: ANYTOWN
Tech State/Province: AP
Tech Postal Code: A1A1A1
Tech Country: AA
Tech Phone: +1.1235551234
Tech Phone Ext: 1234
Tech Fax: +1.5555551213
Tech Fax Ext: 93
Tech Email: EMAIL@EXAMPLE.TLD
Name Server: NS01.EXAMPLE-REGISTRAR.TLD
Name Server: NS02.EXAMPLE-REGISTRAR.TLD
DNSSEC: signedDelegation
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of WHOIS database: 2009-05-29T20:15:00Z <<<

[7] Note: “EXAMPLE” represents one or more TLDs.

Appendix C. Limiting the IDN Variant Domain Names with the Delegation of IDN Variant TLDs

Summary

Variant code points in Label Generation Rules (LGR) for top-level, second-level and other levels can generate variant domain names. Security and Stability Advisory Committee (SSAC) notes that this may introduce a "permutation issue", possibly creating a large number of variant domain names, which "presents challenges for the management of variant domains at the registry, the registrar and registrant levels." Therefore, SSAC advises that "ICANN should ensure that the number of strings that are activated is as small as possible." The current document looks at the factors causing numerous variant labels and suggests measures to address this issue.

The report presents data showing that using the "blocked" disposition for labels addresses the issue to a large extent but not completely. However, the Integration Panel notes that "other steps in the registration process are expected to include suitable mechanisms to shortlist the set of labels for delegation." Therefore, the report looks into multiple reasons which can cause over-production of variant labels. Based on the reasons, it is suggested that domain names may be constrained by using the following measures:

IDN variant TLD labels:

i. Specify for a specific language community
ii. Validate using the relevant language-based Reference Second Level LGR or the relevant language-based second-level IDN table proposed
iii. Determine if usable with generally available input method editors (IME)
iv. Determine if follow the orthographic conventions of the script
v. Demonstrate if meaningful in relevant cases
vi. Consider additional policy to propose a ceiling value and input from Generation Panels

Second-level labels:

i. Encourage use of language-based IDN tables for registration
ii. Determine if based on IDN tables which include code points on principles in RFC 6912, variant code points with types to maximize blocked variant labels, and label-level rules to further reduce the valid or allocatable labels
iii. Block variant labels not contained in a single language-based IDN table
iv. Minimize second-level labels in cases of free or automatic activation
v. Consider additional policy to propose a ceiling value

Domain Names

i. Consider additional constraints when combining top- and second level variant labels
ii. Encourage to have a consistent top-level and second-level policy
iii. Consider additional policy to propose a ceiling value
iv. Include such recommendations for second level in IDN Implementation Guidelines
v. Promote similar practices for the third and other levels, as applicable

Numerosity of IDN Variant Domain Names and their Impact

When variant code points are defined in an IDN table or as part of the Label Generation Rules (LGR) for generating domain labels, they allow for creation of additional labels against the applied-for or primary label by an applicant. For example, using the variant code point sets given in the Root Zone LGR for Arabic script, the new gTLD ابوظبي forms 80 variant labels, as listed at the end of this appendix. All or a subset of such variant labels can be made available to the applicant to activate in the DNS. In some cases, such a subset may cause numerosity of domain names, as described by the Security and Stability Advisory Committee (SSAC) in Recommendation 14 of [SAC060], stating:

> Variants introduce a permutation issue both at the top level as well as with combinations of top level and second level:
>
> - At the TLD level, assume a TLD string with four characters, where each character has three variants. Thus the variant set created would be 3^4= 81 different strings. The size of the variant sets can grow exponentially.
>
> - At 2LD level, assume a 2LD string with four characters, where each character has three variants, and the same number for top level. Thus the variant set created would be 3^4 x 3^4= 72171.

SSAC goes on to explain that the numerosity of domain names may result in problems:

> Such large number of variant strings presents challenges for the management of variant domains at the registry, the registrar and registrant levels. We have seen that some registries have imposed additional rules for variants … Conservatism is also to be used in this case for the root as well.

SSAC concludes that "ICANN should ensure that the number of strings that are activated is as small as possible."

Therefore, the numerosity of variant domain names in the DNS needs to be managed. This permutational issue would be managed by addressing generation of variant labels at all the different levels of the domain names, including the top-level and the second-level. The current document looks at the factors that may cause numerous variant labels at the different levels to suggest measures which may be possible to address this potential issue at each of these levels, to control the permutations for the fully qualified domain names.

Reducing the IDN Variant Top-Level Domain Labels

Using IDN Variant TLD Label Disposition

The Root Zone Label Generation Rules (RZ-LGR) are being developed encompassing the different scripts and writing systems. In addition to the code points which can be used from a

script to form labels, the RZ-LGR also defines variants of these code points as determined by the relevant script community. If a label uses one or more of the code points which have variant code points, it results into generating corresponding variant labels.

To contain the possibility of too many variant labels being generated for the top-level labels, the Procedure to Develop and Maintain the Label Generation Rules for the Root Zone in Respect of IDNA Labels (the LGR Procedure) stipulates to set each variant relation between two code points to either "allocatable" or "blocked" type. When a label is formed using code point(s) which have variants, all the variant labels created containing at least one code point variant with blocked type are given an overall blocked label disposition, and only the remaining variant labels are given the allocatable label disposition. The LGR Procedure specifies that blocked variant labels will not be delegated, whereas the allocatable variant labels may be considered for this purpose. Therefore, the LGR Procedure requires (i) maximizing the blocked variant labels, to promote security of the system by reducing user confusion[8], and (ii) minimizing the allocatable variant labels, to ensure the manageability of the DNS. For example, by designing the RZ-LGR for the Arabic script on these specifications results in generating 79 additional variant labels for ابوظبي (Abu Dhabi) gTLD label, out of which 78 are blocked and only one variant label is allocatable, as shown at the end of this appendix.

Limitations of Using IDN Variant TLD Label Disposition

The variant labels which are eventually delegated will be a subset of all the allocatable variant labels generated by the RZ-LGR, though theoretically all allocatable labels can eventually be delegated. Even though the RZ-LGR is designed to keep the allocatable variant labels small, there can always be limiting cases where many allocatable variant labels could be generated for a particular applied-for label. For example, the table below shows the results of testing done on 476 valid test labels given by the Arabic Generation Panel using the Arabic RZ-LGR covering multiple languages using the Arabic script and arbitrarily covering some common words which can be potentially used as domain labels in these languages. The table shows that though on average only two variant labels are produced for these test labels, one of the labels generates 23 allocatable variant labels (the maximum for these test labels).

Category of Labels	Labels Generated	Average against Total Base Labels	Maximum Variant Labels Against a Label
Valid Base Labels	476	1	-
Allocatable Variant Labels	1033	2	23
Blocked Variant Labels	100844	203	12388
Invalid Labels due to Whole Label Evaluation Rules	7368	15	1200
total	**109721**	**221**	

[8] "Confusability is a security concern," SAC 089: https://www.icann.org/en/system/files/files/sac-089-en.pdf.

This is also acknowledged by the Integration Panel (in charge of publishing the RZ-LGR), which states, as part of the release of RZ-LGR-1, that:

> There are limitations to what can be done with mechanical application of rules, and in some cases, it is not possible to reduce the number of allocatable labels that is practicable and safe without creating undue restrictions on otherwise valid labels. In this context it is a useful reminder that having a label that is "allocatable" neither means that it will necessarily be delegated, nor that it necessarily should be delegated. In fact, investigations of actual registrations on the second level reveal that applicants have tended to apply for only a small number of variant labels.
>
> The LGR can be thought of as creating a maximal set of valid labels and allocatable variants, but other steps in the registration process are expected to include suitable mechanisms to shortlist the set of labels for delegation. It is the view of the Integration Panel that such shortlisting is absolutely necessary, because increasing numerocity of delegated variant is concomitant with an increased risk to the DNS.

If the allocatable variant labels are numerous for certain TLDs, the advice for conservatism in SAC60 by SSAC remains applicable due to the management issues these may cause, as discussed in the previous section. As RZ-LGR only mechanically or algorithmically determines allocatable variants, these could only be further limited for delegation through a policy which stipulates a separate subsequent evaluation process to choose suitable labels from among the allocatable variant labels.

Parameters of Additional Process to Reduce IDN Variant TLD Labels

Mechanism of addressing this issue of over-production of allocatable variant TLDs is already stipulated in the User Experience Report for IDN Variant TLDs, and endorsed as a potential mechanism for further analysis by SSAC. In Recommendation 8 in [SAC060], SSAC states that "***A conservative process needs to be developed to activate variants from allocatable variants in LGR***" (emphasis added). It explains that:

> Based on the SSAC's understanding, given the following LGR calculation: LGR(string) -> string1{state1}, string2{state2}, ..., stringN{stateN} Where state1, state2, ..., stateN is one of the two possible states: allocatable or blocked. A string that is allocatable does not imply automatic activation; rather that it can be allocated. If the string is allocated it is done so "in sync" with the base string that was the input to the LGR. As it is ICANN's role to stipulate this policy, a clear process needs to be developed to avoid ad hoc treatment of new gTLD applications. The user experience report recommends that ICANN must implement a well-defined and conservative variant TLD allocation process. The SSAC agrees with the recommendations below:

- The approval of a variant TLD must not be automatic, but initiated upon the request of a TLD applicant, explicitly specifying (1) the variant label; (2) the status for which the variant should be evaluated (activated, allocated but not activated, etc.); and (3) the need for the variant (e.g., motivated by linguistic, security, usability and/or other considerations). Unless such an application is initiated, all variants generated against a primary TLD application by the root LGR should remain withheld (and unallocated).

In summary, a conservative approach should be taken to delegate any of the allocatable IDN variant TLDs. Such an approach should require the TLD applicant to explicitly apply for any variant TLD label and such a request should be motivated by and evaluated against the linguistic and usability considerations.

Causes of Over-Production of Variant TLDs

Implicit in the argument presented above is that not all allocatable variant labels are equally desirable and if one can determine a mechanism to differentiate them, it may help in determining how their possible delegation may be constrained. To determine how the sub-set of allocatable variants which could be delegated may be determined, it is useful to consider the different ways extraneous allocatable labels may be generated from the usability perspective, with the caveat that the reasons may vary across different scripts.

Difference in the Level for Analysis vs. Use of Code Point Variants

The foremost reason of over-production of allocatable variant labels may be the difference in the context of analyzing variant labels for RZ-LGR compared to the context of their use by end-users. To serve the global community, RZ-LGR is specifically designed in the context of script or writing system, so the generation panel looks at the wider use across all languages and across all communities using the relevant script. Code point repertoire, the variant code points and the label evaluation rules are based on this wider context. However, an applicant of a TLD label will need to look through the lens of a particular language community to be served when identifying a TLD label or its variant label, which may be a much narrower scope for certain scripts. For example, for the design of RZ-LGR for the Arabic script, the community has considered the use of the script for scores of languages spoken in multiple regions, including South-East Asia, South Asia, Central Asia, Middle-East, North Africa and Sub-Saharan Africa. However, a particular TLD label or its variant label will be used by the applicant to target a specific language community, e.g. speakers of Persian language limited to Iran, or speakers of Urdu limited to India and Pakistan, etc.

This may cause at least three possible redundancies in the allocatable variant TLD labels generated:

(i) A variant TLD is supported by the relevant language, but is not the preferred version based on the end-user requirement, (e.g., شبکة is the relevant transliteration but the community also considers شبکہ and شبگہ equivalent);

(ii) A variant TLD is in a language not targeted by the applicant (e.g., شبکة is the Urdu variant label which could be used in India and Pakistan but for the Middle-East and North Africa the Arabic language label شبكة is relevant); and
(iii) The variant TLD is not supported by any language using the script, being an artifact of the script-based analysis (e.g., شبکہ may not be possible to type using a single keyboard, mixing Arabic language Kaf and Urdu language Heh letters).

In case of (i) there may not be a strong argument based on need because the alternate variant TLD label is likely a preference of style of a user. Thus, it may not present a strong case for delegation.

In case of (ii), if the TLD operator intends to target the additional linguistic community, the relevant variant TLD may require delegation. However, in this case, the language (and the locale) would need to be identified to justify the delegation request. To ensure that the TLD variant label is usable for the language, it could be checked that the variant label is valid based on the relevant language-based IDN table submitted for use under the variant TLD label (or as an alternate check the label is valid using the Reference Second Level LGR released by ICANN for that language, if available). If a label cannot be generated by the relevant language-based second level IDN table supported by the variant TLD label, it may not be an appropriate choice for delegation.

However, it should also be noted that if a variant TLD can be created by a language based IDN table, that may in itself may not provide sufficient justification to support it. Additional usability measures, e.g. existing input method, may also be considered. Explicit consideration of the language by the relevant Generation Panel in the script based RZ-LGR proposal and support of the language in the Common Locale Data Repository could also be additional measures which can be considered.

The labels generated due to case (iii) should not be delegated. The RZ-LGR should ideally not generate allocatable IDN variant TLD labels for the cases in (iii), but the algorithmic solution may not always be optimal. A good reason to prevent delegation of such labels is that there would not be a common input method like usual keyboard for the targeted community to generate them; and it would not be expected that users switch between keyboards to type a domain label.

Use of Same Script across Different Writing Systems

Variant code points may be over-generated in cases where the same script is used across different writing systems, for example, use of Han script for Chinese, Japanese and Korean writing systems. Chinese use of Han script requires certain code points to be considered variants of other. The same characters are also used by Japanese writing system. However, Japanese users may consider such code points unique and not variant code points. For example, 学 (U+5B66), 孝 (U+6588) and 學 (U+5B78) are variant code points in Chinese but are considered distinct in Japanese. The RZ-LGR requires a cohesive treatment of variant code points across all the three writing systems for the Han script. So the RZ-LGR will create variant labels due to requirements for Chinese, but such labels will be completely distinct for the

Japanese users, and will be considered redundant and over-produced in the context of Japanese. Though the current RZ-LGR is looking into managing such discrepancies by making such labels blocked for Japanese, some over-production may still be possible.

Usage Conventions

Scripts generally have usage conventions, which limit the use of various code points in certain contexts. For example, generally in all languages using the Arabic script, the Arabic Letter Teh Marbuta ۃ (U+0629) can only occur at the end of a word. This code point is considered variant of another code point, Arabic Letter Heh ہ (U+0647) which can occur in the middle or beginning of a word. Similar example exists in Greek script, where lower-case σ (U+03C3) is replaced by lower-case ς (U+03C2) in word-final position. If these are declared allocatable variant code points, these may create allocatable variant TLD labels which may not be well-formed based on the conventions of the script.

Similarly, many Abugida scripts have formal structures. For example, a dependent vowel always follows a consonant, a tone always follows a consonant or an explicit dependent vowel, etc. Script covered by many Generation Panels, including Neo-Brahmi, Thai, Lao, Khmer, Tibetan, are examples of such cases.

RZ-LGR captures these contextual or structural constraints using label level rules. However, such label level rules may not be optimal in some cases, as these rules are engineered to avoid too much complexity in their design and to cater to multiple languages written in a script simultaneously. Thus, these rules may also allow some allocatable variant labels in some cases which are not well-formed. Being well-formed could be considered as a criterion for selecting from allocatable labels for delegation.

Meaningfulness of Variant TLD Labels

Though not applicable across all domain labels, especially some gTLDs which may be arbitrary and meaningless strings, certain categories of labels, including brands, geographic names, community names, country codes, etc., do require some degree of association with entities or meaningfulness.

Consider the following variant labels generated from the IDN ccTLD for Pakistan using the Arabic script RZ-LGR. Out of the 1200 variants generated, the following six are allocatable or valid and the rest 1194 are blocked. Even from the six allocatable variant labels, the labels in red in the table below (nos. 2, 4, 5, 6) do not represent formal or correct spellings of the country in any language.

#	Label	Disposition	Code Point Sequence	Reason
1	پاکستان xn-- mgbai9azgqp6j	valid	U+067E (پ) U+0627 (ا) U+06A9 (ک) U+0633 (س) U+062A (ت) U+0627 (ا) U+0646	Official name and spelling in Urdu language

			(ن)	
2	پاکستان xn-- mgbai9a5eva00b	allocatable	U+067E (پ) U+0627 (ا) U+0643 (ك) U+0633 (س) U+062A (ت) U+0627 (ا) U+0646 (ن)	Not possible to type using any keyboard – Arabic letters U+067E (پ) and U+0643 (ك)
3	پاکستان xn-- mgbai9azgqpjk	allocatable	U+067E (پ) U+0627 (ا) U+06AA (ک) U+0633 (س) U+062A (ت) U+0627 (ا) U+0646 (ن)	Spellings of Pakistan in Sindhi language
4	پاکستاں xn-- mgbai9az3atike	allocatable	U+067E (پ) U+0627 (ا) U+06A9 (ک) U+0633 (س) U+062A (ت) U+0627 (ا) U+06BA (ں)	Spellings for possible poetic use in Urdu language
5	پاکستاں xn-- mgbai9a5e3r9n	allocatable	U+067E (پ) U+0627 (ا) U+0643 (ك) U+0633 (س) U+062A (ت) U+0627 (ا) U+06BA (ں)	Not possible to type using any keyboard – Arabic letters U+067E (پ) and U+0643 (ك)
6	پاکستاں xn-- mgbai9az3azi5d	allocatable	U+067E (پ) U+0627 (ا) U+06AA (ک) U+0633 (س) U+062A (ت) U+0627 (ا) U+06BA (ں)	Incorrect spellings for Sindhi

Thus, in such contexts, the lack of association or meaningfulness due to variation in spelling of allocatable variant labels may be considered as their over-production and such labels should not be delegated.

Policy Considerations for Reducing IDN Variant TLDs to be Delegated

As discussed allocatable variant labels may be over-produced for many different reasons. These reasons can be considered to determine some mechanism to reduce those which could be delegated, for consideration during the policy and procedure development for this purpose. Again, the motivation is to keep the process conservative by determining the subset of allocatable IDN variant TLDs based on their usability from the perspective of the target language community.

1. The IDN variant TLD label must be specified for a specific language community, which should have been supported by the relevant script Generation Panel in their proposal and is present in the CLDR.
2. The IDN variant TLD for the target language should be valid label using that language-based Reference Second Level LGR released by ICANN, if available. In case the reference LGR is not available or the applicant requires additional characters to support the language, the IDN variant TLD label should be valid using the relevant language-based second-level IDN table proposed for it.

3. The variant label must be usable. People of the target community should be able to compose the variant TLD using generally available input method editors (IME).
4. The variant label should follow the orthographic conventions of the script and the relevant language community.
5. The variant label must demonstrate association or meaningfulness in relevant cases (country or geographic names, brands, trademarks, etc.).

Just to ensure that the number of delegated variant labels remains small, even if with the constraints are not able to adequately limit the subset of label which can be delegated, a ceiling value could also be proposed. For example, the Chinese community allows for three variant labels, one which has been applied-for, one which is the Simplified Chinese version and one which is the Traditional Chinese version. A similar limit may be imposed in the beginning to ensure conservatism, and which may be relaxed over time, as the community gains experience with IDN variant TLD delegation and associated usability and manageability challenges.

Finally, it may be useful to ask the script communities already formed as Generation Panels to list any additional set of recommendations that can be used by evaluation process to consider the appropriateness of the allocatable label for delegation. Based on the policy determined, the applicant should be asked to explain the motivation and need for requesting each variant label of an IDN TLD for delegation and the motivation be evaluated at the time of application based on the criteria finalized.

Reducing the IDN Variant Labels at the Second Level

IDN variant labels at the second level may be created using the applicable IDN tables and the associated policy of the relevant registry. As discussed earlier, SSAC has identified that: "Variants introduce a permutation issue both at the top level as well as with combinations of top level and second level" [SAC060]. Therefore, to address this permutation challenge, the variant labels at second level also need to be limited.

Some of the same mechanisms proposed for the top-level may also be applicable for the second level.

Language-based IDN tables (instead of script-based language tables) should be used where possible. Such IDN tables should be designed to include code points based on the principles in RFC 6912, should provide the variant code points using "blocked" and "allocatable" types maximizing blocked variant code points to address confusability and minimizing allocatable variant code points for promoting manageability, and contain label level rules to further contain the allocatable labels produced.

Variant labels which are produced using code points not contained in a single language-based IDN table should be given a blocked disposition, as these may not be usable by any particular linguistic community.

If the registrant has to pay an extra fee for each additional variant label activated, the registrant will request only the labels needed. Even where such fees are low, the registrant may still not activate too many variant labels due to the costs associated with managing them.

The cases where second-level labels are free for activation or are automatically activated will be harder to manage. In such cases the registry must ensure that the number of variant labels is as small as needed. For example, some registries supporting Chinese domain names normally restrict automatic activation to a total of three labels, including applied-for, Simplified Chinese version, and Traditional Chinese version of the labels. Other script communities considering free or automatic activation of variant labels should develop guidelines on how to limit such labels to a small number.

Finally, the community may also consider putting an arbitrary upper limit on the variant labels registered, considering the IDN tables and additional variant allocation and delegation policies it is implementing. Again, such policies may vary across scripts.

Reducing the IDN Variant Domain Names

Combining variant labels from top-level and variant labels from the second-level for creating the domain names can cause a multiplicative effect, even if the labels have been reduced at each level separately. So, the cases where there are variant labels for the TLD with allocatable variant code points in the associated second-level IDN tables need to be more carefully managed. This can be exacerbated in the cases where second-level IDN variant labels are automatically activated. For example, in the case of Chinese, if a TLD has three variant labels delegated (primary, Simplified Chinese and Traditional Chinese: {PTLD, SCTLD, TCTLD}) and another three labels are automatically activated for the second level (SL): {PSLD, SCSLD, TCSLD}, it creates nine possible domain names:

1. PSLD.PTLD
2. SCSLD.PTLD
3. TCSLD.PTLD
4. PSLD.SCTLD
5. SCSLD.SCTLD
6. TCSLD.SCTLD
7. PSLD.TCTLD
8. SCSLD.TCTLD
9. TCSLD.TCTLD

This may generate too many domain names to manage by the registrant, especially if these are automatically activated. Registration policies need to be considered and extended with additional constraints to manage such consequences. For example, the usability argument could be extended to reduce domain names in Chinese by limiting them to be either all-Simplified or all-Traditional Chinese across the whole domain name, in addition to the applied-for primary label, reducing to only three variant domain names from the nine possible ones in most cases:

1. PSLD.PTLD
2. SCSLD.SCTLD
3. TCSLD.TCTLD

Additional restrictions may also be considered at the time of review of the application for IDN variant TLDs. In case the algorithmic solution and policy constraints are insufficient, having a ceiling value, may be a final way to contain the domain name permutations. The community needs to deliberate on what could be a good starting value and then relax it over time based on experience.

It may be considered to include such recommendations in IDN Implementation Guidelines, which are part of the contractual obligation for the gTLD registries and registrars and recommended for implementation for the IDN ccTLDs through the Fast Track process.

Similar practices may also be carried to the third and other levels, as applicable.

Summary of Recommendations

In summary, the community should deliberate on developing appropriate policy and procedures to manage the permutational challenge caused by introduction IDN variant labels. This requires a three-tier approach: (i) minimizing delegation of IDN variant TLD labels, (ii) minimizing registration of IDN variant labels created at the second level, and (iii) minimizing the domain registrations possible by combining available variant labels at multiple levels. The following measures are recommended for the community to consider for this purpose.

Top-Level Label

1.1. The IDN variant TLD label must be specified for a specific language community, which should have been supported by the relevant script Generation Panel in their proposal and is present in the CLDR.
1.2. The IDN variant TLD for the target language should be valid label using that language-based Reference Second Level LGR released by ICANN, if available. In case the reference LGR is not available or the applicant requires additional characters to support the language, the IDN variant TLD label should be valid using the relevant language-based second-level IDN table proposed for it.
1.3. The variant label must be usable. People of the target community should be able to compose the variant TLD using generally available input method editors (IME).
1.4. The variant label should follow the orthographic conventions of the script and the relevant language community.
1.5. The variant label must demonstrate association or meaningfulness in relevant cases (country or geographic names, brands, trademarks, etc.).
1.6. Community should discuss if additional policy may be developed to propose a ceiling value to ensure that the number of delegated variant TLD labels remains small, even if with the constraints above are not able to adequately limit the subset of allocatable labels which can be delegated, e.g., three labels, as used by the Chinese community.

The community may also consider taking further input from the script-based Generation Panels who have developed the RZ-LGR proposals for this purpose.

Second-Level Labels

1.7. Language-based IDN tables (instead of script-based language tables) should be used for registration where possible.

1.8. Such IDN tables should designed to include the following:

i. Code points, based on the principles in RFC 6912.

ii. Variant code points, using variant code point types which resolve to either "blocked" and "allocatable" dispositions for all valid labels as per RFC 7940, maximizing blocked variant labels.

iii. Label-level rules to further reduce the valid or allocatable labels generated.

1.9. Variant labels which are produced using code points not contained in a single language-based IDN table should be given a blocked disposition.

1.10. For the cases where second-level labels are free for activation or are automatically activated, it should be additionally ensured that the number of activated variant labels is as small as possible. Script-based communities considering free or automatic activation of variant labels should develop guidelines on how to limit such labels to a small number.

1.11. Community should discuss if additional policy may be developed to propose a ceiling value to ensure an arbitrary upper limit on the variant labels registered at the second level, considering the IDN tables and additional variant allocation and delegation policies, e.g., three variant labels, as used by the Chinese community.

Domain Names

1.12. Registration policies need to be considered and extended with additional constraints to manage consequences of combining variant labels at top-level and second level.

1.13. The registration policy being applied for second level should be consistent with the top-level. For example,

1.14. Additional restrictions, such as putting a ceiling value, should be considered in case the algorithmic solution and policy remain insufficient to contain the domain name permutations. For example, an overall limit of three variant domain names could be proposed, as practiced by the Chinese community. Such limits could be script dependent. The community needs to deliberate on what may be a good starting value and then relax it over time.

1.15. It may be considered to include such recommendations for the second level in IDN Implementation Guidelines.

1.16. Similar practices may also be promoted for the third and other levels, as applicable.

Variant Labels of ابوظبي (Abu Dhabi) in Arabic Script

U-label	Disposition	Code point sequence
ابوظبي xn--mgbca7dzdo	valid	U+0627 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+064A (ي)
Variant labels (including original as last)		
80 variant label(s) generated. By disposition: Counter({'blocked': 78, 'allocatable': 1, 'valid': 1})		
ابوظبئ xn--lgbbda3fte	blocked	U+0627 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
ابوظبى xn--mgbca7dzdi	blocked	U+0627 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
ابوظبٻ xn--mgbca7dzdxp	blocked	U+0627 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+067B (ٻ)
ابوظبى xn--mgbca7dzd84b	allocatable	U+0627 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CC (ی)
ابوظبى xn--mgbca7dzdv5b	blocked	U+0627 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CD (ۍ)
ابوظبي xn--mgbca7dzdu6b	blocked	U+0627 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D0 (ې)
ابوظبے xn--mgbca7dzd66b	blocked	U+0627 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)
ابؤظبي xn--jgbfdb3f7e	blocked	U+0627 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+064A (ي)

ابؤظبئ xn--jgbebeb9g	blocked	U+0627 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
ابؤظبى xn--jgbfdb3f1e	blocked	U+0627 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
ابؤظبٮ xn--jgbfdb3fxu	blocked	U+0627 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+067B (ٮ)
ابؤظبى xn--jgbfdb3f89b	blocked	U+0627 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CC (ى)
ابؤظبى xn--jgbfdb3fv0c	blocked	U+0627 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CD (ۍ)
ابؤظبي xn--jgbfdb3fu1c	blocked	U+0627 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D0 (ې)
ابؤظبے xn--jgbfdb3f61c	blocked	U+0627 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)
آبوظبي xn--hgbma7dzdo	blocked	U+0622 (آ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+064A (ي)
آبوظبئ xn--hgbifa3fte	blocked	U+0622 (آ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
آبوظبى xn--hgbma7dzdi	blocked	U+0622 (آ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
آبوظبٮ xn-- hgbma7dzdxp	blocked	U+0622 (آ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+067B (ٮ)
آبوظبى xn-- hgbma7dzd84b	blocked	U+0622 (آ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CC (ى)
آبوظبى xn-- hgbma7dzdv5b	blocked	U+0622 (آ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CD (ۍ)
آبوظبي xn-- hgbma7dzdu6b	blocked	U+0622 (آ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D0 (ې)
آبوظبے xn--	blocked	U+0622 (آ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)

hgbma7dzd66b		
آبؤظبي xn--hgbelb3f7e	blocked	U+0622 (آ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+064A (ي)
آبؤظبئ xn--hgbeggb9g	blocked	U+0622 (آ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
آبؤظبى xn--hgbelb3f1e	blocked	U+0622 (آ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
آبؤظببٖ xn--hgbelb3fxu	blocked	U+0622 (آ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+067B (ٻ)
آبؤظبى xn--hgbelb3f89b	blocked	U+0622 (آ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CC (ی)
آبؤظبى xn--hgbelb3fv0c	blocked	U+0622 (آ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CD (ۍ)
آبؤظبي xn--hgbelb3fu1c	blocked	U+0622 (آ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D0 (ې)
آبؤظبے xn--hgbelb3f61c	blocked	U+0622 (آ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)
أبوظبي xn--igbka7dzdo	blocked	U+0623 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+064A (ي)
أبوظبئ xn--igbgfa3fte	blocked	U+0623 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
أبوظبى xn--igbka7dzdi	blocked	U+0623 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
أبوظببٖ xn--igbka7dzdxp	blocked	U+0623 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+067B (ٻ)
أبوظبى xn-- igbka7dzd84b	blocked	U+0623 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CC (ی)
أبوظبى xn-- igbka7dzdv5b	blocked	U+0623 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CD (ۍ)
أبوظبي xn--	blocked	U+0623 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D0 (ې)

igbka7dzdu6b		
أبوظبے xn-- igbka7dzd66b	blocked	U+0623 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)
أبوظبي xn--igbclb3f7e	blocked	U+0623 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+064A (ي)
أبوظبئ xn--igbcggb9g	blocked	U+0623 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
أبوظبى xn--igbclb3f1e	blocked	U+0623 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
أبوظبپ xn--igbclb3fxu	blocked	U+0623 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+067B (پ)
أبوظبى xn--igbclb3f89b	blocked	U+0623 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CC (ی)
أبوظبى xn--igbclb3fv0c	blocked	U+0623 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CD (ۍ)
أبوظبي xn--igbclb3fu1c	blocked	U+0623 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D0 (ې)
أبوظبے xn--igbclb3f61c	blocked	U+0623 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)
إبوظبي xn--kgbga7dzdo	blocked	U+0625 (إ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+064A (ي)
إبوظبئ xn--kgbcfa3fte	blocked	U+0625 (إ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
إبوظبى xn--kgbga7dzdi	blocked	U+0625 (إ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
إبوظبپ xn--kgbga7dzdxp	blocked	U+0625 (إ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+067B (پ)
إبوظبى xn-- kgbga7dzd84b	blocked	U+0625 (إ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CC (ی)
إبوظبى xn--	blocked	U+0625 (إ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CD (ۍ)

kgbga7dzdv5b		
إبوظبي xn-- kgbga7dzdu6b	blocked	U+0625 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D0 (ى)
إبوظبے xn-- kgbga7dzd66b	blocked	U+0625 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)
إبؤظبي xn--jgbbjb3f7e	blocked	U+0625 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+064A (ي)
إبؤظبئ xn--jgbbegb9g	blocked	U+0625 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
إبؤظبى xn--jgbbjb3f1e	blocked	U+0625 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
إبؤظبٻ xn--jgbbjb3fxu	blocked	U+0625 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+067B (ٻ)
إبؤظبى xn--jgbbjb3f89b	blocked	U+0625 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CC (ی)
إبؤظبى xn--jgbbjb3fv0c	blocked	U+0625 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CD (ۍ)
إبؤظبي xn--jgbbjb3fu1c	blocked	U+0625 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D0 (ى)
إبؤظبے xn--jgbbjb3f61c	blocked	U+0625 (ا) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)
أبوظبي xn-- ngba1c7cm9u	blocked	U+0672 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+064A (ي)
أبوظبئ xn-- lgbda7dzdwm	blocked	U+0672 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
أبوظبى xn--ngba1c7ch5v	blocked	U+0672 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
أبوظبٻ xn-- ngba1c7c2k5b	blocked	U+0672 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+067B (ٻ)

أبوظبى xn--ngba1c7c2k5t	blocked	U+0672 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CC (ى)
أبوظبى xn--ngba1c7c2kku	blocked	U+0672 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06CD (ى)
أبوظبي xn--ngba1c7c2k2u	blocked	U+0672 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D0 (ى)
أبوظبے xn--ngba1c7c2knv	blocked	U+0672 (أ) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)
أبؤظبي xn--jgbhb7dtezk	blocked	U+0672 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+064A (ي)
أبؤظبئ xn--jgbeeb3fwr	blocked	U+0672 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0626 (ئ)
أبؤظبى xn--jgbhb7d6dvl	blocked	U+0672 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+0649 (ى)
أبؤظببٻ xn--jgbhb7d9o3b	blocked	U+0672 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+067B (ٻ)
أبؤظبى xn--jgbhb7d9ows	blocked	U+0672 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CC (ى)
أبؤظبى xn--jgbhb7d9o2s	blocked	U+0672 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06CD (ى)
أبؤظبي xn--jgbhb7d9ort	blocked	U+0672 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D0 (ى)
أبؤظبے xn--jgbhb7d9o3t	blocked	U+0672 (أ) U+0628 (ب) U+0624 (ؤ) U+0638 (ظ) U+0628 (ب) U+06D2 (ے)
ابوظبي xn--mgbca7dzdo	valid	U+0627 (ا) U+0628 (ب) U+0648 (و) U+0638 (ظ) U+0628 (ب) U+064A (ي)