DEVANĀGARĪ VIP TEAM ISSUES REPORT

DEVANĀGARĪ VIP GROUP

Contents

0. PRELIMINARIES

a. Background and Overview

Thanks to the policy of opening up scripts other than Latin by ICANN, a flood-gate of new languages and scripts has opened up and domain-names will become truly multi-lingual in nature. Benefiting from this new policy, India has taken up the challenge of providing IDN's in Indian scripts and languages for the 22 official languages of Indian (see Appendix I). The formulation of a policy document for India to provide Internationalized Domain Names in the 22 official languages has been nearly 5 years in the making. Started in 2005, the policy has been elaborated over the years to ensure that the eventual users will have as safe as an environment as possible when they register their names in an Indian language using their native script.

7 Indian languages (Hindi, Tamil, Telugu, Gujarati, Bangla, Urdu and Punjabi) have already been proposed to ICANN and IANA and the ccTLD for the country name "India" in these languages have already been approved and delegated into the DNS root zone.

Since scripts do not share the same composition rules and have their own "grammar of composition"; it was in the fitness of things, that ICANN felt that the creation of "test cases" in six scripts would allow for a better perception of the problems as well as issues involved. The scripts chosen for study (apart from Latin): Greek, Cyrillic, Arabic, Devanāgarī , Chinese reflect in fact the 4 major writing systems of the world Abugidas (Greek and Cyrillic), Abjads (Arabic), Akshar or Alphasyllabaries (Devanāgarī) and Phonetic-Semantic (Chinese).

Within this perspective a series of discussions via e-mail were initiated. A team was constituted for Devanāgarī (cf. Appendix I) which embraced not only Hindi but other major languages using the Devanāgarī script (cf. Appendix II). The discussions culminated in a meeting of all the groups at Singapore in June and another meeting of the Devanāgarī group at Pune in July.

Over a series of discussions both prior to the creation of the case-study team and after, a slow consensus building process has been evolving and a major step towards this process is a preliminary draft in which each script delineates its problems, issues especially with reference to its writing structure and the notion of variants arising there from.
It is these concerns and issues which this report addresses. The report attempts to lay down the background to writing system along with the various issues for the creation of Internationalized Domain Names in Languages using Devanāgarī. It is the result of discussions, teleconferences, email exchanges as well as document formalizations over the past months in order to arrive at a working draft which is proposed in what follows.

b. Structure

The report, whose basic layout was finalized at a meeting the case study team held in Pune , comprises the following sections:

Part 1 attempts to set things in perspective by providing an overview of the evolution of Devanāgarī, the languages that use Devanāgarī and also a brief sketch of the writing system of the language.

Since the aim of this document is to highlight issues pertinent to all aspects of IDN variants: linguistic, technical, societal, fiscal, and administrative, these issues are highlighted in a sequential order[1]. Part 2 is an inventory of the major issues pertinent to the topic in question and examines the problems from all angles.

Since the Registry plays an important role in IDN, a special section, Part 3 is devoted to this area.

A certain number of Appendices which provide ancillary information and also treat of the issues of definitions and questions raised at the Singapore meet, complete the report.

[1] Since some of these are interesting but do not have direct relevance to the issue of Variants, they have been listed in Appendix V

1. DEVANĀGARĪ : AN OVERVIEW

This over-view of Devanāgarī is a linguistic introduction to Devanāgarī . It starts off with the historical evolution of Devanāgarī and in section 1.2 studies the structure of Devanāgarī . Section 1.3 develops the notion of the underlying nucleus: the akshar and further draws attention to certain akshar structures relevant to variants. IPA as well as simple transliteration has been used as a guide to the pronunciation of the examples.

1.1 Devanāgarī: A Historical Perspective

Devanāgarī (pronounced [d̪eːʋˈnaːɡriː])is the main script for the Indo-Aryan languages Hindi, Marathi, Maithili and Nepali recognized as official languages of the Republic of India. It is the only script also for the related Indo-Aryan languages Bagheli, Bhili, Bhojpuri, Himachali dialects, Magahi, Newari and Rajasthani. It is associated closely with the ancient languages Sanskrit and Prakrit. It is an alternative script for Dogri, Kashmiri (by Hindu speakers), Sindhi and Santali. It is rising in use for speakers of tribal languages of Arunachal Pradesh, Bihar and Andaman & Nicobar Islands. Devanāgarī can be easily shown to be related to the modern scripts used for other Indian languages such as Gujarati, Gurumukhi (for Punjabi), and Assamese/ Bengali, as well as to the scripts used for Dravidian languages, such as Tamil, Telugu, Kannada and Malayalam.

It is now well-known that Devanāgarī has evolved from the parent script Brāhmī, with its earliest historical form known as Aśokan Brāhmī , traced to the 4th century B.C. Brāhmī was deciphered by Sir James Prinsep in 1837. The study of Brāhmī and its development has shown that it has given rise to most of the scripts in India, as mentioned above, and some outside India, namely, Sri Lanka, Myanmar, Kampuchea, Thailand, Laos, and Tibet.

The evolution of Brāhmī into present-day Devanāgarī involved intermediate forms, common to other scripts such as Gupta and Śāradā in the north and Grantha and Kadamba in the South. Devanāgarī can be said to have developed from the Kutila script, a descendant of the Gupta script, in turn a descendent of Brāhmī. The word *kutila,* meaning ‘crooked’, was used as a descriptive term to characterize the curving shapes of the script, compared to the straight lines of Brāhmī. A look at the development of Devanāgarī from Brāhmī gives an insight into how the Indic scripts have come to be diversified: the handiwork of engravers and writers who used different types of strokes leading to different regional styles (cf..Singh 2006).

In spite of the diversified character of Brāhmī-derived scripts, they have a common structure. An understanding of the structure of Devanāgarī , or for that matter of any of the scripts derived from Brāhmī, is of general interest for this group of scripts of South and Southeast Asia.

1.2 The structure of written Devanāgarī

The writing system of Devanāgarī could be summed up as composed of the following:

1.1.1. The Consonants

Devanāgarī consonants have an implicit schwa /ə/ included in them. As per traditional classification they are categorized according to their phonetic properties. There are 5 (Varg) groups and one non-Varg group. Each Varg contains five consonants classified as per their properties. The first four consonants are classified on the basis of Voicing and Aspiration and the last is the corresponding nasal.

Varg	Unvoiced		Voiced		Nasal
	-Asp	+Asp	-Asp	+Asp	
1 Velar	क	ख	ग	घ	ङ
2 Palatal	च	छ	ज	झ	ञ
3 Retroflex	ट	ठ	ड	ढ	ण
4 Dental	त	थ	द	ध	न
5 Bi-labial	प	फ	ब	भ	म

Non-Varg

य	र	ल	ळ	व	श	ष	स	ह

1.1.2. The Implicit Vowel Killer: Halanta[2]

All consonants have an implicit vowel sign (schwa) within them. A special sign is needed to denote that this implicit vowel is stripped off. This is known as the Halanta (◌्) . The Halanta thus joins two consonants and creates conjuncts which can be from 2 to 3 consonant combinations (cf. 1.2. supra)

1.1.3. Vowels

Separate symbols exist for all Vowels which are pronounced independently either at the beginning or after a vowel sound. To indicate a Vowel sound other than the implicit one, a Vowel modifier (Mātrā) is attached to the consonant. Since the consonant has a built in schwa, there are equivalent Mātrās for all vowels excepting the अ.

The correlation is shown as under:

अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ
	◌ा	ि◌	◌ी	◌ु	◌ू	◌ृ	◌े	◌ै	◌ो	◌ौ

[2] Unicode (cf. Unicode 3.0 and above) prefers the term Virama. In this report both the terms have been used to denote the character that suppresses the inherent vowel.

In addition to show sounds borrowed from English, some languages using Devanāgarī such as Hindi, Marathi, and Konkani also admit 2 vowels and their corresponding Mātrās as in
ऍ ॅ ऑ ॉ

ऍण्ड /and/ ऑर /or/

Marathi replaces the ऍ by ॲ

1.1.4. The Anuswāra /ं/ represents a homo-organic nasal. It replaces a conjunct group of a Nasal consonant+Halanta+Consonant belonging to that particular varg. Before a Non-varg consonant the anuswāra represents a nasal sound. Modern Hindi, Marathi and Konkani prefer the anuswāra to the corresponding Half-nasal:

सन्त vs. संत /sənt/ saint   चम्पा vs. चंपा /tʃəmpa/

1.1.5. Nasalization: Chandrabindu ँ

Chandrabindu/Anunasika denotes nasalization of the preceding vowel as in आँख (eye) /ãkh/ eye. Present-day Hindi users tend to replace the chandrabindu by the anuswāra

1.1.6. Nukta़[3]

Mainly used in Hindi, the nukta sign is placed below a certain number of consonants to represent words borrowed from Perso-Arabic. It can be adjoined to क ख ग ज फ to show that words having these consonants with a nukta are to be pronounced in the Perso-Arabic style.
e.g. फ़िरोज़ /firoz/

It is also placed under ड ढ in Hindi to indicate flapped sounds

With the exception of flaps, users of modern-day Hindi hardly use the nukta characters today

1.1.7. Visarg ः and Avagrah ऽ

The Visarg is frequently used in Sanskrit and represents a sound very close to /h/. दुःख /du:kh/ sorrow, unhappiness

The Avagrah ऽ creates an extra stress on the preceding vowel and is used in Sanskrit texts. It is rarely used in other languages using Devanāgarī.

1.3. This classification of Devanāgarī characters can be reduced to a "compositional grammar" based on a Backus-Naur formalism (ISCII '91) which ensures the well-formedness of the akshar. The term used in this report ABNF (Augmented Backus-Naur Formalism) refers to the fact that apart from (L) Letters the formalism will also handle (H) Hyphen and (D) Digits.

[3] The nukta will be treated at length in the section of Normalization, since Unicode allows the characters mentioned above to be represented in two different ways: as a single character or a consonant+the nukta

1.3 The Fundamental Unit: akshar

The *akshar* is the graphemic unit of Devanāgarī. The difference between the syllable and the akshar is that while the syllable includes one or more post-vocalic consonants, the akshar doesn't, as can be seen below:

Phonemic forms	Syllabic units	Akshara units
cha:ruləta:	CV. CV. CV. CV	CV. CV. CV. CV
e:k	VC.	V. C
upka:r	VC. CVC	V. C. CV. C
indira	VC. CV. CV	VC. CV. CV
əst	VCC	V. CC
əkʃər	VC. CVC	V. CCV. C

Table 1: Syllabic and akshara divisions of spoken forms

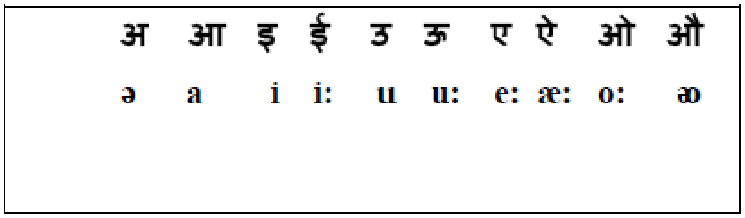
As can be seen from table 1, there is a marked difference between the written and spoken syllable, especially insofar as the division of consonant clusters across syllable boundaries e.g. /upka:r/ is concerned.

The only exception to the generalization about the post-vocalic consonants vis-à-vis akshar is the anuswāra, the underlying nasal consonant surfacing as homorganic with the following stop. The anuswāra is treated as a part of the grapheme. The orthographic and phonetic transcriptions of forms with the anuswāra are given below:

बिंदी	[bindi:]	'point$_N$'
कंबल	[kəmbəl]	'blanket$_N$'
डंडा	[ɖəɳɖa:]	'stick$_N$'
खंजर	[k^həɲɟər]	'knife$_N$'
कंघी	[kəŋ$g^ɦ$i:]	'comb$_N$'

Table 2: Representation of anuswāra in Devanāgarī

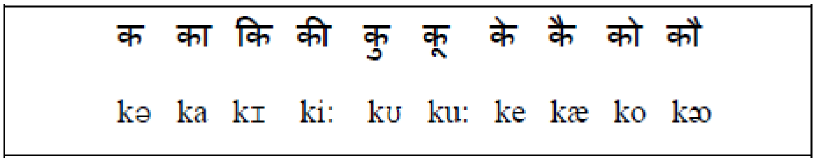
1. A vowel is an independent unit of *akshar* word-initially and post-vocalically.



अ	आ	इ	ई	उ	ऊ	ए	ऐ	ओ	औ
ə	a	i	i:	u	u:	e:	æ:	o:	aɔ

Table 3: Independent vowel letters

a. Vowels and consonants are assumed to be different types of units and are so represented in the grapheme when the vowels follow consonants. The following akshar consist of single consonants followed by a vowel:



क	का	कि	की	कु	कू	के	कै	को	कौ
kə	ka	kɪ	ki:	kʊ	ku:	ke	kæ	ko	kaɔ

Table 4: *Devanāgarī CV akshar*

2. As can be seen in the first grapheme in Table 3, the neutral vowel /ə/ is assumed to be inherent in a consonant. The vowel is pronounced as such word initially and medially in certain contexts, for example, in the first grapheme in पल /pəl/. The inherent neutral vowel is not pronounced word-finally or medially in certain contexts.

Two-consonant clusters

5. Generally, half the letter of the first consonant precedes the full letter of the second consonant: e.g., स्क <sk> , प्त <pt>, क्ल <kl> etc. Alternatively, the practice of specifying the diacritic for unreleased consonants, known as 'halanta', is used for the first consonant, e.g., द् भ<db^{h}> उद् भव/udb^{h}əʋ/

6. For a C+r cluster, as noted above, the /r/ is specified as a subscript that looks like an inscript: क्र <kr>, ख्र <k^{h}r>, फ्र <p^{h}r>.

7. For r+C clusters, the the /r/ is specified as a superscript above the grapheme, e.g., र्म <rm>, र्ज <rʤ>

8. In the case of the following two-consonant clusters, a new ligatured group is formed. These are: त्र <tr>, क्ष <kʂ>, ज्ञ<ɟɲ>, श्र <ʃr>, क्त <kt>.

Three-consonant clusters:

9. Generally, the first two consonants are specified for half their letters, and the third is fully specified, e.g., स्प्ल <spl>. This convention is usually followed for borrowed words.

10. For C+C+r clusters, and for r+C+C clusters, which are highly restricted, the convention for two-consonant clusters applies, e.g., स्त्र <str>

2. ISSUES

From a typological point of view, the following practical considerations need to be taken into account when discussing issues and trying to identify solutions:

a. ccTLD's vs. gTLD. While the former are under the control of a policy determined by a given country, the latter do not fall within the compliance of such a policy

b. Introduction of the notion of language tables, restriction rules and well-formedness constraints (in Brāhmī derived languages) and variant-hood to reduce spoofing, pharming and phishing. Thus for Brāhmī based languages which are akshar driven, a formalism needs to be evolved to handle well-formedness.

c. Potential areas where such factors apply. These are:

   1. Issues arising out of the possible implementation of ZWJ/ZWNJ as prescribed in IDNA 2008.
   2. Issues relating to required Devanāgarī characters that are not Protocol Valid.
   3. Issues arising out of software behavior, particularly in relation to how domains are displayed.
   4. Issues arising out of Registry Management. Issues specific to management of certain top-level domains.

These will be developed in what follows. By way of conclusion a tabular summing-up of issues has been provided.

2.1 Language vs. Script Issues

Within the ccTLD for .भारत the dichotomy of language vs. script issues can be handled (with certain issues to be tackled at the registry level) , but at the gTLD level, it is assumed that only script will dominate which will require the adoption of new strategies for handling issues pertinent to language especially variants.

2.2 Variants in Devanāgarī Script

Variants in Indian Languages as defined in the Indian policy for .भारत are based either on similarity obtained through normalization or visual look-alikes (limited to consonant clusters, it being assumed that single characters do not lend themselves to spoofing). Five types of variants can be identified. Of these the first two are because of Unicode issues and the third is a true set of visual variants based on visually confusing characters. The variants of type 4 (c.f. 2.2.4) and 5 (c.f. 2.2.5) have not been considered under .भारत policy but have been mentioned here considering the broader scope of issues for this case study team

2.2.1 Variants generated from legacy input methods

Earlier versions of Unicode did not have certain characters. In order to generate these characters alternative methods such as the use of Halanta followed by a ZWJ (U+200D) were used.
e.g. Eye-lash ra[4]

र्‍ U+0930 U+094D U+200D	ऱ् U+0931 U+094D

Unicode 2.0 prescribes the use of RA+VIRAMA+ZWJ to represent the eyelash-ra. This is captured in what was then rule R5 of Section 9 (which is now rule R5a). Unicode 3.0/4.0 reflected the ISCII choice, in what is now rule R5: "In conformance with the ISCII standard, the half-consonant form rrah is represented as eyelash-ra. This form of ra is commonly used in writing Marathi…" (Unicode 3.0)

So, the word दऱ्या/ darya/ "valleys5 can be written with the Unicode values U+0926 **U+0930 U+094D U+200D** U+092F U+093E (दर्‍या) as well as U+0926 **U+0931 U+094D** U+092F U+093E (दऱ्या)

2.2.2 Variants generated because of Combining Characters

These variants exist because Unicode allows for two or more ways of representing certain characters. Unicode handles the issue through Normalization.
Thus in the case of Devanāgarī the "nukta" character is the candidate for Normalization . e.g. a sample of two such instances is provided.

क+◌़ U+0915 U+093C	क़ U+0958
ख+◌़ U+0916 U+093C	ख़ U+0959

As per revised IDNA standard, "IDNA 2008" the atomic form of nukta characters have been marked as "disallowed", still as a precautionary

[4] The eyelash ra is used in Konkanai, Nepali and Marathi. Denoted as र् it is treated as different from the र् (repha) by certain linguists. While the former is treated as a flap, the latter is a continuant trill (*cf.*, Kalyan Kale and Anjali Soman. 1986). There are cases in Marathi of minimal pairs such as: आचार्यास "to the teacher" vs. आचाऱ्यास "to the cook or दर्या /darya/ "ocean" vs. दऱ्या /darya/ "valleys. Similar cases may exist in Konkani and Nepali.

[5] The stand-alone shape of eyelash ra र्‍ cannot be shown due to rendering issues.

measure, they have been mentioned as variants and have also been kept within the ambit of the policy for .भारत ccTLD.

2.2.3 Confusingly similar shapes

2.2.3.1. Single characters

These are the characters which have confusingly similar shapes. However, this category of variants were not considered in the .भारत ccTLD policy as there was a possibility that this approach would result in barring many useful domain names from being registered.
e.g.

घ U+0918	ध U+0927
भ U+092D	म U+092E

Table 4

This table contains only a sample list. A full-list is provided in Appendix IV

2.2.3.2. Composite characters

These are conjuncts that look alike and can be easily confused in the small URL bar of the browser. These look-alikes have been identified for each language.
e.g.

द्ग U+0926 U+094D U+O917	द्र U+0926 U+094D U+0930	द्न U+0926 U+094D U+0928
द्ध U+0926 U+094D U+0927	द्घ U+0926 U+094D U+0918	
ष्ट U+0937 U+094D U+091F	ष्ठ U+0937 U+094D U+0920	
द्व U+0926 U+094D U+ 0935	द्ब U+0926 U+094D U+092C	

Table 5

This table contains only a sample list.

2.2.4 Cross-script character mixing

There has been a possibility to allow mixing of scripts within a label in certain top-level domains, especially gTLD's[6]. Our opinion is that the introduction of cross-script characters is extremely dangerous and could result in spoofing, phishing and scamming. The policy for .भारत ccTLD does not allow code block mixing.

Assuming that such cross-script character mixing will be for gTLD's, a list of cross-lingual visual similarities is provided below. It should be noted that such similarities are restricted to single characters and not to conjuncts. Spoofing can be possible by mixing characters from these different code blocks.

DEVANĀGARĪ **SCRIPT**	**COGNATE** **SCRIPT**	**CODE POINT IN COGNATE SCRIPT**
VOWELS		
उ U+0909	Bangla	ও U+0993
उ U+0909	Gurmukhi	ਤ U+0A24
ऋ U+090B	Gujarati	ૠ U+0AE0
CONSONANTS		
क U+0915	Bangla	ক U+0995
ग	Gujarati	ગ

[6] Cf. Section 4 *IDN Variant TLDs - Lists of Issues - v06 redline[1]* which talks of cross-script character mixing

U+0917		U+0A97
ग U+0917	Gurmukhi	ਗ U+0A17
घ U+0918	Gurmukhi	ਬ U+0A2C
घ U+0918	Gujarati	ઘ U+0A98
ङ U+0919	Gujarati	ઙ U+0A99
छ U+091B	Gujarati	છ U+0A9B
ञ U+091E	Gujarati	ઞ U+0A9E
ट U+091F	Gurmukhi	ਟ U+0A17
ठ U+0920	Gujarati	ઠ U+0AA0
ठ U+0920	Gurmukhi	ਠ U+0A20
ड	Gujarati	ડ

U+0921		U+0AA1
ढ U+0922	Gurmukhi	ਢ U+0A2B
त U+0924	Gujarati	ત U+0AA4
ध U+0927	Gujarati	ધ U+0AA7
न U+0928	Gujarati	ન U+0AA8
न U+0928	Bangla	ন U+09A8
न U+0928	Bangla	ণ U+09A3
प U+092A	Gujarati	પ U+0AAA
प U+092A	Gurmukhi	ਧ U+0A17
प U+092A	Gurmukhi	ਪ U+0A2A
प	Gurmukhi	ਪ

U+092A		U+0A6B
म U+092E	Gurmukhi	ਸ U+0A38
म U+092E	Gujarati	મ U+0AAE
य U+092F	Gujarati	ચ U+0A9A
र U+0930	Gujarati	ર U+0AAE
र U+0930	Gurmukhi	ਕ U+0A15
ल U+0932	Bangla	ল U+09B2
व U+0935	Gujarati	વ U+0AB5
श U+0936	Gujarati	શ U+0AB6
श् U+0936 U+094D	Bangla	ঽ U+09BD
ष	Gujarati	ષ

U+0937		U+0AB7
स U+0938	Gujarati	સ U+0AB8
ह U+0939	Gujarati	હ U+0AB9
Nukta characters		
ग़ U+095A or U+0917 U+094D	Gurmukhi	ਗ਼ U+0A5A
ढ़ U+095D Or U+ 0922 U+094D	Gurmukhi	ਫ਼ U+0A5E

Table 6

2.2.5 Homophonic Variants

In Devanāgarī based languages, homophonic variants which admit two homophones (spelling variants as in English *color-colour*) e.g. हिंदी and हिन्दी /hĩdi:/[7] do occur but the rules for such variants are ill-defined and could increase the chances of malfeasance. Within the ambit of the ccTLD policy for .भारत such variants have not been considered .

2.3 Issues Pertaining to Unicode Normalization

While Unicode does provide rules for normalization which are reflected in IDNA2008, a major issue arises:

[7] cf. 1.3 supra

Within Unicode itself a large number of normalizations are not defined. Although such instances of missing normalizations do not occur in Devanāgarī, the area needs considerable exploration and with the continuous enrichment of characters to the Devanāgarī Code Block the chances of such missing normalization increase..

e.g. Devanāgarī character U+094E Devanāgarī vowel sign Prishthamatra E (used in Vedic) could be used along with a ZWJ to look like U+093E Devanāgarī vowel sign AA for purposes of spoofing

◌ॎ U+094E	◌ा U+093E

A similar case is that of the eye-lash ra in Devanāgarī where Unicode provides two possible input methods but does not treat them as a case of canonical normalization (cf. 2.2.1 supra)
Similar instances are possible in other code-blocks such as Urdu (0600) and Malayalam (0D00) which, although not within the purview of this report which treats of Devanāgarī, have been quoted as a matter of interest.

Arabic code block U+600:

ڈ U+0688	◌ؕ د U+062F U+0615

A similar case occurs in Malayalam written in Malayalam script.

ന്‍ U+0D28 U+0D4D U+200D	ൻ U+0D7B

2.4 Zero Width Joiner (ZWJ) and Zero Width Non-Joiner (ZWNJ) :

ZWJ (U+0200D) and ZWNJ (U+0200C) are code points that have been provided by the Unicode standard to instruct the rendering of a string where the script has the option between joining and non-joining characters. Without the use of these control codes, the string may be rendered in an alternate form from what is intended. This is mostly applicable to those forms which are alternatives of each others. In each case the use of ZWJ is specified and the issues arising out of the said use are provided next

2.4.1 Zero Width Joiner (ZWJ)

The ZWJ plays multiple roles.

2.4.1.1 Used to generate half form of base consonant in “Base-Cons+Halanta+Cons”

There are some cases of conjunct formation in languages written in the Devanāgarī script in which the basic shapes of two characters being joined by Halanta are not retained. If in such cases if the conjunct form in which the basic shapes(in some cases as half forms) of the combining characters is to be retained, the ZWJ is used after Halanta.

e.g.

क (ka) + ◌् (halanta) + ष (ssha) -> क्ष (kssha)

क (ka) + ◌् (halanta) + ZWJ + ष (ssha)-> क्‍ष (k+ssha)

Issue :

The issue that arises in this usage of ZWJ is that, there are some conjuncts which by default are represented in the form where the basic shapes (in some cases as half forms) of the combining characters are retained. In such cases the use of ZWJ after Halanta character does not make any difference visually. Thus we eventually get two strings which have different storage but same visual appearance.[8]

e.g.

क (ka) + ◌् (halanta) + न (na) -> क्न (kna)

क (ka) + ◌् (halanta) + ZWJ + न (na) -> क्‍न (kna)

Our observation has shown that even a skilled human being cannot disambiguate these two. Constraint rules can be written to handle this issue, by identifying which combination of two consonants with and without ZWJ yields the same visual results. However this is partly offset by the fact that the shape which is formed by combining characters is highly dependent on font and/or underlying rendering engine. (cf. Browser Issues infra)

[8] This and the parallel problems with ZWNJ, are potentially serious. The contextual rule provided in IDNA2008 is intended to be fairly generic and to isolate issues to the point where registry restrictions are sufficient. It would not be surprising if the actual registry rules would be required to be more extensive and specific in order to avoid difficulties. From that perspective, these examples are simply confirmation that more sophistication is needed in rules about what should be permitted to be registered.

Though this behavior is largely governed by the language needs, there are still some cases where discrepancies are observed and thus such cases cannot be clearly identified and singled out.

2.4.1.2 To generate certain special characters

To generate out some characters in Indian Languages including Devanāgarī based languages, Unicode provided a combination with the use of ZWJ. e.g. in Marathi which is a Devanāgarī based language to generate out “eyelash ra” (cf. 2.2.1. for a discussion on the same)
र (ra) + ् (halanta) + ZWJ -> र्‍ (eyelash ra)

Issue :

The issue that arises in this case is, two different combinations:

र्‍ U+0930 U+094D U+200D	ऱ् U+0931 U+094D

will result in same visual form . Including this kind of combination in variant table will solve this issue.

2.4.2 Zero Width Non-Joiner (ZWNJ: U+200C))

ZWNJ on the other hand is used, to put in broad sense, to explicitly display virama between two characters which otherwise would have joined to form a conjunct. As per Unicode (Chapter 9) “*Explicit Virama (Halant). Normally a virama character serves to create dead consonants that are, in turn, combined with subsequent consonants to form conjuncts. This behavior usually results in a virama sign not being depicted visually. Occasionally, this default behavior is not desired when a dead consonant should be excluded from conjunct formation, in which case the virama sign is visibly rendered. To accomplish this goal, the Unicode Standard adopts the convention of placing the character U+200C zero width non-joiner immediately after the encoded dead consonant that is to be excluded from conjunct formation. In this case, the virama sign is always depicted as appropriate for the consonant to which it is attached.*”

e.g.
क (ka) + ् (halanta) + ष (ssha) -> क्ष (kssha)
क (ka) + ् (halanta) + ZWNJ + ष (ssha) -> क्‌ष (k+ssha)

In the latter case, we can see the combining characters retaining their forms, with the halanta which is a joining character, having explicit visual appearance.

Issue :

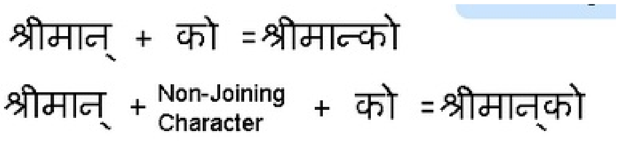
1. The issue that arises in this usage of ZWNJ is that, there are some conjuncts which by default are represented in the form where the halanta has explicit visual appearance even in the absence of ZWNJ. In such cases the use of ZWNJ after Halanta character does not make any difference visually. Thus we again

eventually get two strings which have different storage but same visual appearance[9].

e.g.
ड (dda) + ◌् (halanta) + द (da) -> ड्द
ड (dda) + ◌् (halanta) + ZWNJ + द (da) -> ड्‌द

Similar to the case of ZWJ, the shape which is formed by combining characters is highly dependent on font and/or underlying rendering engine. Though this behavior is largely governed by the language needs, there are still some cases where discrepancies are observed and thus such cases cannot be clearly identified and singled out. A study of such forms needs to be undertaken to detect if constraint rules can handle this issue.

2. In languages such as Nepali, the use of ZWNJ permits the correct generation of certain Noun paradigms, as illustrated in the following example:



श्रीमान् + को =श्रीमान्को
श्रीमान् + Non-Joining Character + को =श्रीमान्‌को

The word श्रीमान् ends in a Virama. Adjoining to it the suffix को generates an incorrect form where the suffix and the root form a conjunct श्रीमान्को. This would be inacceptable to the user community. To ensure that the root form and the suffix are clearly indicated, ZWNJ is inserted as shown in the example above.
Constraining rules cannot be applied in this case since the number of such paradigms is very large. The choice is to admit ZWNJ (and also possibly the risk of malfeasance or not admit such forms)

2.5 Issues relating to required Devanāgarī characters that are not Protocol Valid

2.5.1. Case of U+02BC

The character U+02BC *Modifier Letter Apostrophe* / ʼ/ which acts as a tone mark or length mark is used very frequently in languages like Boro, Dogri, Maithili

[9] Since the comprehensive list of such combinations is voluminous, it is provided separately with the report as a set of PDF files.

which are Devanāgarī script based and Bangla which is Bengali script based. An example from Dogri where 02BC is used as a syncopation marker will clarify the issue:

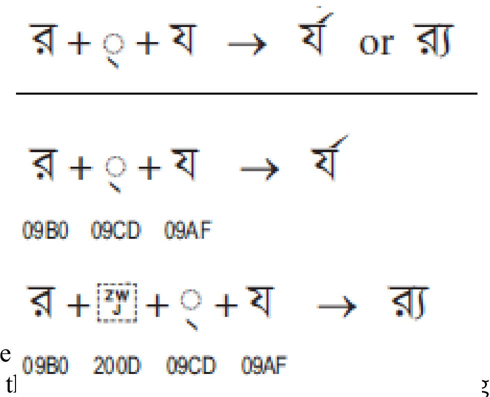
कराʼरदा । (means : got done)

Issue :

U+02BC *Modifier Letter Apostrophe* character comes from the block U+02B0-U+02FF,whereas all the characters which belong to Devanāgarī script come from the block U+0900-U+097F. If as a policy decision, script mixing is not allowed in gTLDs, this character still be allowed as an exception because without this character the language representation will not be complete[10]. It may be noted that the keyboards devised for languages (Boro, Dogri, Maithili) using this character provide the means of entering the character which has a relatively high frequency of usage in these languages.

2.5.2. Use of ZWJ (U+200D)

As per IDNA 2008 protocol, the ZWJ has been permitted with the restriction that the preceding character must be a "virama". In languages using the Devanāgarī script, ZWJ is used to display some combinations with same set of combining characters but different visual appearance. Though this case does not exist in languages written in the Devanāgarī script (at present)[11], it is found in other Indian language scripts. This is mentioned precisely to reinforce the point that Unicode has proposed the use of ZWJ in a manner which is not valid as per IDNA2008. This leads to the concept that it would be better to identify such cases (few and far-between) and devise constraint rules to handle the same.
The case of *"Interaction of Repha and Ya-phalaa"* which exists in Bengali script is a prime example. In general in Indian languages, the combination of "ra+halanta" when followed by a consonant generates a "repha". In case of Bengali, the combination "halanta+ya" is called as "ya-phala". When this combination is preceded by "ra" an ambiguous situation arises. Unicode[12] has proposed, that ZWJ be inserted after "ra" (which is not a *virama*) to generate ra with ya-phala.



[10] The case needs to be debated since

[11] This "coda" has been added since t[illegible] ɡarī is constantly being enriched by additions

[12] Chapter 9. Unicode 6.0 http://unicode.org/

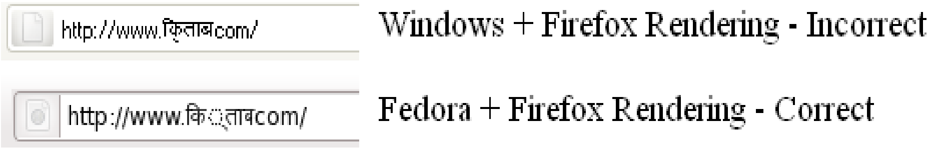
2.6 Issues Related to Software Behavior in Relation to Display of Domains :

The DNS is not exclusively about the web but also affects other areas such as email user agents, calendaring programs, etc. However as a case study, issues pertaining to browsers have been taken up. The issues highlighted here are applicable to other software behavior in relation to display of domains .

The browsers for representing the domain name in the URL bar of the browser, rely on the underlying OS rendering engine. Thus the issues associated with the rendering engines of the OS are inherent in the browser. The fonts that get applied on the URL bar in IDNs are chosen by the browsers as per default font for the script of the domain name provided by the underlying OS.
The issues related to these characteristics of the browsers belong to two broad categories as

1 Rendering Engine related issues
2 Font related issues

1. Rendering Engine related issues :
Whenever some text is submitted to a Unicode Enabled application, the rendering engine breaks this text in the form of syllables. These syllable formation rules have not been standardized, nor has Unicode given any specific rules pertaining to the same. Thus the behavior of different rendering engines is different and depends on the understanding of the language/script of the implementing body which seldom is perfect. This is exemplified in the theoretical cases given below which show how under different environments the same browser does not display/displays a mal-formed syllable:



http://www.कि्ताबcom/

Windows + Firefox Rendering - Incorrect

http://www.कि्ताबcom/

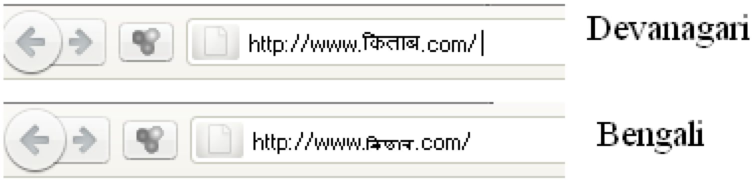
Fedora + Firefox Rendering - Correct

The theoretical example given above with a valid label: किताब (book) shows how the rendering engine of the operating system permits mal-formed syllables. The test domain entered is किताब /kitāb/with a halanta/virama after the first syllable: कि. Firefox under Windows shows that this incorrect syllable is not rendered as malformed. The same under Fedora shows up the malformation of the syllable and is hence marked as "correct" whereas the first instance is flagged as "incorrect".

2. Font related issues :
In case of rendering of Domain Names in browsers, font that gets applied on the domain name in address bar of the browser plays major role. Each operating system has a specific font which act as a default font for every script/language

the OS supports. The browser uses default font provided by the OS for displaying the domain name in the address bar.

Similar to the rendering engine, the font implementation also varies from vendor to vendor. And thus the same Domain Name can be seen differently depending on the font properties, orthography adopted by the font, hinting, weight, kerning etc as can be seen in the example below where Hindi and Bengali in the same point size have different visual display: Hindi being more readable than Bengali.



http://www.किताब.com/

http://www.কিতাব.com/

As there is no central authority that can ensure consensus implementations, it is hoped that a user-facing applications software that claims to support Devanāgarī should have a listed set of capabilities would go a long way toward improving and rationalizing the user experience.

2.7 Issues arising out of Registry Management

Assuming that all other factors and conditions are satisfied, a major issue touches upon the registry. Registry issues can be divided in to the following parts.

A major caveat to this discussion is that a policy for a given script (Devanāgarī in this instance) can apply only to top ccTLD and the second-level domain. The third level and subsequent level domains do not fall within the ambit of the policy. A theoretical example will illustrate the case:

Given www.किताब.भारत. The policy developed (for India in this case) would apply only to किताब. The sub-domains would not be governed by the policy. Thus in the case of www.पुस्तक.किताब.भारत, the policy which would apply to किताब may not apply to पुस्तक. A consensus on this issue needs to be taken and appropriate mechanisms need to be evolved or a call needs to be taken by respective registries as to the "depth" to which the policy evolved will apply.[13]

Given this major caveat, the following issues arise out of registry management

2.7.1. Delegation of variant TLD's

2.7.2. Email Addresses resolution

[13] Domain hacks, where the second-level domain and ccTLD are used together to form one word or one title. This has resulted in domains like blo.gs of South Georgia and the South Sandwich Islands (gs), del.icio.us of the United States (us), and cr.yp.to of Tonga (to) are a good instance of such cases. (Example taken from Wiki) Similar domain hacks are possible for Devanāgarī , especially at the third-level where the policy does not apply at all..,

2.7.1. Delegation of variant TLD’s

There is a strong possibility that the zone generation process might be affected when variants of a given TLD label are supplied to it. This eventuality raises certain issues which need serious consideration:

1. Identification of the variant type. In the case of Devanāgarī based scripts, different variant types have been identified (cf. 2.2 above). The Registry will have to interact differently with each variant type. Variants which require normalization and those exist because Unicode has permitted over its evolution two or more input methods for representing the same character will need to be handled differently from visual look-alikes.
2. Corollary to the above is the question of how the zone file for a given TLD variant be handled ? Will it share the same zone file or will allocation be made in the registry for each variant? Basically the registry will have to take a call and as mentioned above, accommodate the variant as per its variant type[14].
3. The final issue is that of language and script. Given that a script supports more than one language (Devanāgarī and Bengali in the case of Brāhmī based languages) how should the registry handle this problem in terms of resource records ? Should for example a TLD admitting a variant in Nepali be pertinent to domains appertaining to that language alone ? In the present state there is no way to implement the dichotomy script/language in the absence of any available language tags.
4. Finally as a digression of 3. above, outside the ambit of a ccTLD i.e. gTLD how will the disambiguation function across language and script? In the present state, script seems favored over language. Once more it needs to be stated that there is no way to implement such a disambiguation in the absence of any available language tags.

2.7.2. Email Addresses resolution

The queries raised here are pertinent to .भारत but could also apply in certain instances to other registries.

1. The problems raised in 2.4(supra) have a marked resolution for resolution of email addresses. Given an email such as

   वित्त-मंत्रालय.भारत: Ministry of Finance

   Will the owner of the address also inherit the variant

   वित-मंत्रालय.भारत

[14] From a technical point of view, there are two ways a variant could work: 1. Delegate another zone. 2. Use a DNAME on the parent side to redirect the entire tree. (Comment provided by Mr. Andrew Sullivan)

given that within the .भारत registry त्त U+0924 U+094D U+0924 generates out त U+0924 as a variant

वित्त U+0935 U+093F **U+0924 U+094D U+0924**	
वित U+0935 U+093F **U+0924**	

2. In case both emails are valid, will there be an aliasing mechanism ?
3. The issue is also closely tied with that of the mail-server resolving the email.

2.8. Administrative Issues [15]

These issues are pertinent to the policy to be adopted by the Government of India in the domain of opening up domain names, reserved names, conflict resolution and also the fee structure.

Certain issues arise here, quite a few of which are in the nature of legalities and economic policies.

2.8.1. RESERVED NAMES LIST

Reserved names Lists are deployed for sensitive names which need to be protected by a given country. In the case of .भारत, the following issues could arise, especially with regard to gTLD's:

1. Would gTLD's need a reserved list? Will the Government send a list of reserved names of political sensitivity ? If so are payment issues involved ? (in which case specific processes would be needed for variants)
2. Should all variants of a given gTLD be also requested for blocking ?

2.8.2. DISPUTE RESOLUTION

This is an area of legal policies and mechanisms need to be evolved for handling the same, especially given the introduction of multi-lingual labels. While areas such as "bad faith" and cyber-squatting" already have legal redress mechanisms multi-lingualism brings in its own issues: Multi-lingual dispute claims. These are bundles containing labels in different languages. The following major issues can be identified here:

1. How does a complainant claim rights to a whole label ?

2. Can a complaint be filed if a complainant comes to know that a party has filed for a domain name in which the complainant has valid claims
3. Decision-making mechanisms
   Are precedents allowed ? And if so what mechanism will be evolved for precedents ?

[15] Although not truly within the purview of the Variant Issues Project, the issues presented here could widen the debate and are hence retained.

Would a separate set of mechanisms need to be involved in multi-lingual ownership?
An important issue is that of expertise in resolving a dispute. Simply put who will deem a complaint as valid in the area of a multi-lingual dispute. Will the matter be referred to the State Government or to a competent language authority [16]?

4. International Trademark resolution:
 Which procedure would be followed when a trademark or domain name is claimed by two countries
e.g. Tamil is shared both by Sri Lanka and India as an official language. What would happen if a trademark in Tamil for a corporate in one country closely resembles a similar one belonging to a corporate in the other country ?
Will the label be frozen and treated subjudice during the period of litigation ?
5. Government vs. an Individual or a Corporate body
Will priority be given to Government over Individual claim in case of such a litigation ?

2.8. 3. PAYMENT ISSUES
With the creation of multi-lingual labels and also variants generated from each, certain issues of payment arise:
Will there be a fee for providing and registering Variants
Will there be a fee for a registrant desirous of removing a variant granted to him (issues of cyber-squatting)
Will there be a concession for providing the registrant a label in multiple languages ?

2.9. Management Of Multi-Lingual[17] gTLD's

The issues raised here are specific to gTLD's where TLD's managed outside a country's law .
Certain issues need to be discussed in this area:

1. How are these to be allocated, especially when more than one country shares the same language.?
2. Will the gTLD's permit code-mixing i.e. permitting more than one script to be used within a given gTLD ?
3. Will there be a specific reservation for a country to register its societal and politically sensitive names such as political parties, name of a language etc ?

[16] The term within the Indian context refers to apex bodies such as the Sahitya Akademi (http://sahitya-akademi.gov.in) or Institutions specially created for development of Indian Languages such as the Central Institute of Indian Languages (www.ciil.org/) which are referred to in matters of Indian languages

[17] Although this section may seem to be out of the ambit of the VIP, it is presented to open up the debate of multi-lingual communities. The discussion is not only with reference to countries having more than one language but more specific to languages shared by more than one country. Thus Bengali is shared by both India and Bangle-Desh,, Tamil is the official language of India, Sri Lanka and Singapore. Hindi is shared by both India and Fiji, Nepali by India and Nepal.

4. And corollary to the above which policy will apply for generation of variants ? Will the registrant be permitted to block out variants which are possible ? What would be the financial implications of the same ?
5. If a given corporate body is desirous of registering a gTLD in a variety of scripts, which policy will apply. It is suggested that the policy determined for each script/language be applied to resolve the issue. Thus for the code block U+600, should the policy adopted by the Arabic study group be applied[18] ?
6. If the above suggestion is accepted, what measures are taken in the case of a script shared by more than one country, in case the given countries have different policies

[18] "…this only works up to a point. Two examples (there are others): for Han (CJK) script, Chinese requires variants for SC<->TC matching while Korean and Japanese do not use variants at all. For the closely-related Greek, Latin, and Cyrillic scripts, any rules that are meaningful would have to address the rather large overlap among those scripts" (Comment of John C Klensin on this issue).

2.10. Summing Up

The following table sums up the above discussion for easy reference:

ISSUES	SUB-ISSUES
Linguistic Issues	Language vs. Script. While the ccTLD for .भारत the dichotomy can be handled, at the gTLD level, only script will dominate which implies adopting new strategies for handling variants
Unicode Normalization issues	In the case of Brāhmī-based Scripts (Devanāgarī script is the test case) as well as Scripts derived from the Arabic Code block (U+600) , there is an urgent need to study possible normalization rules not covered by Unicode and by IDNA2008.
Issues arising out of the possible Implementation of ZWJ/ZWNJ as prescribed in IDNA 2008	ZWJ can be handled by constraint rules. Such rules will need to be defined as far as possible. ZWNJ for generating noun paradigms for languages like Nepali need to be discussed since there is no rule-governed behavior.
Issues relating to required Devanāgarī characters that are not Protocol Valid	Case of Boro, Dogri, Maithili which use a character from the Spacing Modifer code block: U+02BC / ' / and which cannot be accommodated in the present conditions laid down by IDNA2008
Issues related to software behavior in relation to display of domains	1. Rendering Engine related issues 2. Font related issues
Issues arising out of Registry Management	Delegation of variant TLD's Email Addresses resolution
Issues specific to gTLD's	How are these to be allocated? Will the gTLD's permit cross-script mixing i.e. permitting more than one script to be used within a given gTLD ? Will there be a specific reservation for a country to register its societal and politically sensitive names? Which policy will apply for generation of variants ? If a given corporate body is desirous of registering a gTLD in a variety of scripts, which policy will apply ? What measures are taken in the case of a script shared by more than one country, in case the given countries have different policies ?

Table 7

3. REGISTRAR AND REGISTRY PERSPECTIVE

Within a registry, there is an important technical consideration when registering internationalized domain names. The domain name must be tagged with both a script indication and a language indication. In order to achieve this a registry will have to establish certain policies that must be enforced when a request to register a domain name is received. The technical issues to be considered in the development of these policies are as follows.

In some cases, it may be sufficient to tag a domain name with either its script or its language. For example, the Gurumukhi script is only used for the Gurumukhi language. In this case the registry can infer the language when it receives a domain name with the Gurumukhi script tag.

Similarly, only the Tamil script supports the Tamil language. Thus when a domain name is tagged with the Tamil language the registry can infer the Tamil script tag.

However, either the Devanāgarī or Perso-Arabic script can support the Sindhi language. In this case when the registry receives a domain name to be registered it must be tagged with both its language and its script.

Also, the Devanāgarī script can be used to support many languages, e.g., Hindi or Nepali. In this case when the registry receives a domain name to be registered it must be tagged with both its language and its script.

The technical issue is that there is no standard way to do this in the standard EPP protocol used by gTLD registries and those ccTLD registries that choose to follow the ICANN recommendations. There is a defined extension for including each of these values but not both together. This issue is being currently pursued with the IETF.

This issue also affects registrars in two ways. To the extent there is no standard, a registrar will have to implement all EPP extensions that various registries may choose to specify to resolve this issue. For those cold’s that do not use EPP registrars will have to implement whatever is required in order to support that ccTLD.

In addition, when registrars are present they are the interface to the registrant. Registrars that choose to support multiple scripts and languages will need to develop user interfaces that facilitate and simplify the identification of the script and language in use by a registrant.

Finally, with respect to the issue of a preferred variant, our discussions have noted that in general no variant is preferred over any other variant. However, RFC 3743 requires that at least one code point be specified in the preferred variant column of a language table. In the context of the Devanāgarī script it would be preferred if the preferred variant column could be left blank until a registrant chooses the desired code point. At that time, operationally, a registry could then insert the chosen code point in to the preferred variant column before proceeding with the rest of the registration process.

3.1. DNS Technology and Operations Perspective

It is important to keep in mind that the DNS is technically a pure lookup protocol: a request is made for specific information (DNS record type) indexed by a domain name that is returned in a response. In the case of internationalized domain names, the domain name in the request is required to be an A-LABEL. Perhaps more importantly, the DNS is agnostic with respect to language and script as this information is neither stored in the DNS nor directly available in any part of the global DNS infrastructure. In that context, from a purely technical point of view, internationalized domain names do not present any unique challenges to the operation of the DNS.

However, a common point of discussion in the context of internationalized domain name TLDs is the desire to "alias" one TLD with another. The specifics of the desired "alias" behavior are varied but the intent, conceptually, is that a lookup of a domain name in one TLD return the same response as a corresponding lookup in the "aliased" TLD. For the two domain names to be corresponding the intent is usually that they be "variants" of each other, and therein lies the principal point of contention. There is no consensus as to the definition of "variant".

A full treatment of the possible definitions is beyond the scope of this comment. However, it is important to note that not all definitions can be fully implemented and enforced with today's DNS technology. This will have an effect on registry policies regarding "aliasing".

The critical gap is that policies regarding DNS behavior cannot be enforced beyond the level in the DNS hierarchy at which the policy is defined. Specifically, a registry may choose to establish a policy wherein all possible variants will behave the same (return the same response in the DNS) at the TLD level of the DNS hierarchy. Although this can work in many cases at the TLD level, the DNS cannot enforce this policy on the delegated second-level domain names in the TLD. This can have a dramatic affect on the user experience.

Security and Stability

A suggestion for evaluating variant policies and their implementation is to log, review, and analyze DNS query traffic. Specifically, the behavior of applications and services, and sometimes the users that use them, can be inferred from traffic patterns found in sequences of DNS queries and responses. For example, registries could review DNS traffic of the TLD for queries of non-existent domains (i.e., in DNS terms reviewing the NXDOMAIN responses). An analysis of these transactions may indicate that language tables are incomplete or that variant usage is not as expected.

Providing a consistent, uniform, and non-surprising (i.e., user expected) experience to the user is an essential component of stability. DNS transaction logs provide some insight into user expectations and thus some ability to confirm that the needs of a user community are being met.

Some TLDs may wish to consider partnerships with second-level domain holders to continue the analysis at lower levels in the DNS hierarchy.

3.2. User Perspective

There are two issues to be considered from a user's perspective when introducing internationalized domain names: the submission and display of internationalized domain names. There are two underlying technical issues. First, can a user enter the desired Unicode code point in to the system? The answer depends in part on the hardware (does the keyboard in use make the code point available) and also on the software (will the software accept the code point value as a valid entry). Second, will the system in use display the Unicode code point in a way that is recognizable to the user? The answer depends in part on the availability of an appropriate font table indexed by the code point with a value representing a glyph that will be recognized by the user when displayed.

These issues are mostly straightforward to resolve in a local context but, when considered in a global context, they become challenging when you consider how a user is expected to maintain their environment such that it "works" in all cases. In this context, "works" means that the user experience remains uniform and consistent, i.e., the user is not surprised by any entry or presentation event. Specifically, consider the case of a web browser.

Web browsers today are commonly regionally packaged, which means it is possible to obtain a browser for whom its default behavior is optimized for the regional scripts or languages in use. However, this requires that appropriate hardware and software is available to support the browser (and the user). In addition, a user's usage of a browser frequently extends beyond the regional area, which means that a user may encounter web sites or information on web sites (documents) that cannot be displayed or used in the local environment without additional configuration (changes to the hardware or software or both).

The critical question is how the local environment (hardware and software) is maintained in the presence of changing entry and presentation needs or requirements?

3.3. System Administrator Perspective

The system administrator as a role is responsible for maintaining a local environment. In an enterprise situation there is a higher probability of greater skill being present and, thus, the maintenance of the local environment is more likely to be constrained by resources (e.g., staff or money). However, many users have mobile devices or other personal resources for which they serve the dual role of system administrator and end-user. These users are more likely to lack the skills necessary to properly maintain their local environment in order to achieve the best user experience possible.

3.4. End-User Perspective

Registration: It is important to keep in mind that the vast majority of users are monolingual and that in many cases the language and script are not Latin-based. The DNS requirement that queries of internationalized domain names be executed with the A-LABEL form of the name presents a burden for end-users. The A-LABEL form of the name is an encoding that transforms the name (using a reversible mapping) such that it is comprised only of US-ASCII characters. This transformation ensures that the use of internationalized domain names is backwards compatible with the existing DNS infrastructure. Working with the A-LABEL form is a burden for many end-users, in part because the encoding presents itself as a random sequence of US-ASCII characters but primarily because working with it is unnatural, even for those familiar with US-ASCII.

The use of appropriate software can mitigate this burden, the consequence of which is that users are constrained by their local hardware and software.

Access: EDITORIAL NOTE: In retrospect, it is not clear that this element in the overall document structure is needed. Given the introduction proposed above for the User Perspective section we do not have anything to add for this section.

3.5. WHOIS Issues

The critical WHOIS issue facing the deployment of IDNs is the fact that the standard WHOIS protocol (as defined by RFC 3912) has not been internationalized, which means there is no standard way to indicate either a preferred language or script, or the actual language or script in use. The WHOIS protocol is a simple request and response transaction: a domain name is submitted to a server and output is returned. The predominant encoding in use on the Internet today is US-ASCII but a consequence of the lack of internationalization is that there is an increasing number of local, regional, and proprietary solutions that attempt to address the lack of internationalization. This variability has a dramatically adverse effect on the user experience.

For example, the labels used to tag the information in the WHOIS response are predominantly indicated in US-ASCII. It is straightforward to believe that the labels should be show in the same language or script as the data itself, but this is not possible with the standard WHOIS protocol.

Secondary to this issue, the question of what to display is a policy issue that will be guided, in part, by the variant registration policy. Consider the following questions.

1. If a variant domain name exists in the registry database but is not present in the DNS (i.e., the domain name is reserved), should a WHOIS request for the domain name return a referral indicating the name is a variant of a superordinate name or return the response for the superordinate name? Should the response indicate the name does not exist?
2. Should variant domain names be permitted to have different WHOIS information associated with them? The answer to this question should depend in part on whether different owners are permitted to register variant domain names.
3. If a variant domain name is a different language or script than its corresponding superordinate domain name, how is this to be presented to the user if the user does not understand (or perhaps cannot display) the superordinate domain name’s language or script?
4. If a WHOIS request is for a domain name with variants, should the variants be included in the response? What if the language or script of the variants cannot be understood or displayed by the user making the request?

3.6. Registration Process Issues

The critical technical issue facing the registration of IDNs and variants is the fact there is no standard way in the EPP protocol to indicate the language, script, or both in use by a domain name to be registered. As described in the Registry and Registrar perspective, this affects the user interface provided to a registrant as well as a registry's ability to know which domain name among a set of variants to register.

Secondary to this issue, a registry will need to have a policy specifying how it will deal with variants of prospective domain name registrations. Consider the following questions.

1. Are domain name variants to be considered equivalent, for an appropriate definition of equivalence?
2. If variants are equivalent, will all be registered (including DNS delegation) when the first one is presented? Will variants be reserved (does not include DNS delegation) and only registered upon request?
3. If variants are reserved for registration upon request, who is permitted to request registration? The owner of the first registered variant or anyone who requests it?

A critical technical issue to the question of equivalence is the implications to the DNS as described in the DNS Technology and Operations Perspective. The DNS behavior cannot be enforced beyond the level in the DNS hierarchy at which the policy is defined. This can have a dramatic effect on the user experience.

Finally, from a business perspective, a registry will need to have a policy specifying how it will charge (or not charge) for variants of registered domain names.

3.7. DNSSEC Issues

There are no IDN or variant specific issues that affect the deployment of DNSSEC.

From the point of view of DNSSEC, an IDN or variant TLD is simply another zone. Recall from the DNS Technology and Operations Perspective discussion that the DNS has no context with respect to the purpose or value judgment of the labels in a zone. The DNS is technically a pure lookup protocol.

A common point of discussion is to correlate the issue of TLD "aliasing" with the key management issues that must ordinarily be resolved when deploying DNSSEC. This coupling is an unnecessary complexity since the questions related to implementing key management should be answered only in the context of DNS and DNSSEC, i.e., an IDN or a variant should be just a "label" to the DNS and DNSSEC.

4. SELECT BIBLIOGRAPHY

The bibliography given below and sorted thematically is a set of documents, books, articles and webographies consulted in the drafting of this report

WRITING SYSTEMS

Dillinger. D., The Alphabet. A Key to the History of Mankind. 3rd Edition in 2 Volumes. Hutchison. London. 1968.

DEVANĀGARĪ

Agrawala, V. S. (1966). The Devanāgarī script. In: Indian Systems of Writing. (Pp. 12-16) Delhi: Publications Division.

Agyeya, Sacchindanand Hiranand Vatsyayan. 1972. Bhavanti. Delhi: Rajpal and Sons.

Beames, John. 1872-79. A Comparative Grammar of the Modern Aryan Languages of India. 3 vols. London, Trubner and Co. [Reprinted by Munshiram Manoharlal, New Delhi, 1966.]

Bhatia, Tej K. 1987. A History of the Hindi Grammatical Tradition: Hindi-Hindustani Grammar, Grammarians, History and Problems. Leiden/New York: E. J. Brill.

Bright, W. (1996). The Devanāgarī script. In P. Daniels and W. Bright (eds), The World's Writing Systems. (Pp. 384-390). New York: Oxford University Press.

Cardona, George. 1987. Sanskrit. In The World's Major Languages. Bernard Comrie (ed.). London: Croom Helm. 448-469.

Dwivedi, Ram Awadh. 1966. A Critical Survey of Hindi Literature. Delhi:Motilal Banarsidass.

Faruqi, Shamsur Rahman. 2001. Early Urdu Literary Culture and History.Delhi: Oxford University Press.

Guru, Kamta Prasad. 1919. Hindi Vyakaran. Varanasi: Nagari Pracharini Sabha. (1962 edition).

Kachru, Yamuna. 1965. A Transformational Treatment of Hindi Verbal Syntax. London: University of London Ph.D. dissertation (Mimeographed).

Kachru, Yamuna. 1966. An Introduction to Hindi Syntax. Urbana: University of Illinois, Department of Linguistics.

Kalyan Kale and Anjali Soman, 1986. Learning Marathi. Shri Vishakha Prakashan, Pune :

McGregor, R. S. (1977). Outline of Hindi Grammar. 2nd edn. Delhi: Oxford University Press.

McGregor, R. S. 1972. Outline of Hindi Grammar with Exercises. Delhi: Oxford University Press.

McGregor, R. S. 1974. Hindi Literature of the Nineteenth and Early Twentieth Centuries. Wiesbaden: Harrassowitz.

McGregor, R. S. 1984. Hindi Literature from Its Beginnings to the Nineteenth Century. Wiesbaden: Harrassowitz.

Pandey, P. K. (2007). Phonology-orthography interface in Devanāgarī for Hindi. Written Language and Literacy, 10 (2): 139-156. 2007.

Rai, Amrit. 1984. A House Divided. The Origin and Development of Hindi/Hindavi. Delhi: Oxford University Press.

Sharad, Onkar. 1969. Lohiya ke Vicar. Allhabad: Lokbharati Prakashan.

Singh, A. K. (2007). Progress of modification of Brāhmī alphabet as revealed by the inscriptions of sixth-eighth centuries. In P.G. Patel, P. Pandey and D. Rajgor (eds), The Indic Scripts: Paleographic and Linguistic Perspectives. (Pp. 85-107). New Delhi: DK Printworld.

Sproat, R. (2000). A Computational Theory of Writing Systems. Cambridge University Press.

Tiwari, Pandit Udaynarayan. 1961. Hindi Bhasha ka Udgam aur Vikas [The Origin and Development of the Hindi Language]. Prayag: Leader Press.
Verma, M. K. 1971. The Structure of the Noun Phrase in English and Hindi.Delhi: Motilal Banarsidass.

MULTILINGUALISM

GENERIC

Multilingual Internet Names Consortium. MINC.
Dam, Mohan, Karp, Kane & Hotta, IDN Guidelines 1.0, ICANN, June 2003
Martin J. (December 20, 1996). "URLs and internationalization". World Wide Web Consortium. IDN TABLES: http://www.iana.org/domains/idn-tables/

LANGUAGE SPECIFIC

2. *INDIAN SCRIPTS AND LANGUAGES*

IS 10401: 8-bit code for information interchange. 1982
IS 10315: 7-bit coded character set for information interchange. 1985
IS 12326: 7-bit and 8-bit coded character sets-Code extension techniques. 1987
ISO 15919, Information and documentation - Transliteration of Devanāgarī and related Indic scripts into Latin characters. 2001
ISO 2375: Procedure for registration of escape sequences. 2003
ISO 8859: 8-bit single-byte coded graphic character sets - Parts 1-13. 1998-2001
IDN POLICY http://mit.gov.in/sites/upload_files/dit/files/India-IDN-Policy.pdf

Romanisation of Indian scripts

Library of Congress. Romanization Standards.. USA. 2002
Stone. Anthony., http://homepage.ntlworld.com/stone-catend/trind.htm

3. *CHINESE*

CHINESE: Chinese Domain Name Consortium". CDNC. 2000-05-19

4. *URDU*

URDU: http://urduworkshop.sdnpk.org

Romanisation of Indian scripts

RFC's

RFC 2181, Clarifications to the DNS Specification: section 11 explicitly allows any binary string
RFC 2870 Root Name Server Operational Requirements June 2000
RFC 3490 Internationalizing Domain Names in Applications (IDNA) March 2003
RFC 3492, Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA), A. Costello, The Internet Society (March 2003)
RFC 5890 "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework"
RFC 5891 "Internationalized Domain Names in Applications (IDNA): Protocol"
RFC 5892 "The Unicode Code Points and Internationalized Domain Names for Applications (IDNA)" August 2010

RFC 5893 "Right-to-Left Scripts for Internationalized Domain Names for Applications (IDNA)"

UNICODE

Unicode Consortium. Unicode ver.3.0.
---. Unicode ver.3.2.
---. Online version of Unicode ver.4.1 . (archived).
----. Online version of Unicode ver. 5.0 & 5.1. (www.unicode.org)
----. Online version of Unicode ver.6.0 (www.unicode.org)

5. LIST OF APPENDICES

Appendix I: Devanāgarī Team Members.

Appendix II: List of Official Languages of India.

Appendix III: Discussion on the Definitions and Questions proposed at Singapore meet.

Appendix IV: List of single character “look-alikes” in Devanāgarī

Appendix V: Topics extraneous to the Variant Issues Project, but deemed to be of interest.

APPENDIX 1:

Devanāgarī Team Members

Member	Role
Dr. Govind	Case Study Coordinator
Dr. Mahesh Kulkarni	Team Member
K. B. Narayanan	Team Member
Dr. James Galvin	Team Member
Amardeep Singh Chawla	Team Member
Tulika Pandey	Team Member
Jitender Kumar	Team Member
Rajiv Kumar	Team Member
Bhavin Turakhia	Team Member
Shashi Bharadwaj	Team Member
Prof Pramod Pandey	Team Member
Dr. Raiomond Doctor	Team Member
Dr. Kalyan Kale	Team Member
Prabhakar Kshotriya	Team Member
Manish Dalal	Team Member
Basanta Shrestha	Team Member
Bal Krishna Bal	Team Member
Satyendra Kumar Pandey	Team Member
Neha Gupta	Team Member
Akshat Joshi	Team Member

STAFF MEMBERS

Member (staff)	Role
Francisco Arias	Subject Matter Expert (Registry Ops)
Naela Sarras	Case Study Liaison
Nicholas Ostler	Subject Matter Expert (Linguistics)
Steve Sheng	Subject Matter Expert (Policy)
Andrew Sullivan	Subject Matter Expert (Protocol)

APPENDIX II:
List of Official Languages of India[19]

India is a linguist's hunting ground with 4 major language families, over 6616 languages (Census of India 2001) and 20000+ dialects having been identified[20] (SIL report). To face this vast diversity, a considerable amount of accommodation has been made by the Constitution of India which has stipulated the usage of Hindi and English to be the two languages of official communication for the national government. In addition a set of 22 scheduled languages have been identified which are languages that can be

a. officially adopted by different states for administrative purposes,
b. as a medium of communication between the national and the state governments,
c. for examinations at the University as well as government levels.
d. for national databases such as voter lists, Unique Identity Number program (UIDAI) etc.

The 22 scheduled languages are represented table wise as under :

Language	ISO	Official Language	Family	Script
Assamese	asm	Assam	Indo-Aryan	Assamese
Bengali	ben	Tripura and West Bengal	Indo-Aryan	Bangla
Boro	brx	Assam	Tibeto-Burman	Devanāgarī (modified)
Dogri	dgr	Jammu and Kashmir	Indo-Aryan	Devanāgarī (modified)
Gujarati	guj	Dadra and Nagar Haveli, Daman and Diu, and Gujarat	Indo-Aryan	Gujarati
Hindi	hin	Andaman and Nicobar Islands, Bihar, Chandigarh, Chhattisgarh, Delhi, Haryana, Himachal Pradesh, Jharkhand, Madhya Pradesh, Rajasthan, Uttar Pradesh and Uttaranchal	Indo-Aryan	Devanāgarī
Kannada	kan	Karnataka	Dravidian	Kannada
Kashmiri	kas		Indo-Aryan	Perso-Arabic Devanāgarī
Konkani	kok	Goa	Indo-Aryan	Devanāgarī Roman (Latin)
Maithili	mai	Bihar	Indo-Aryan	Devanāgarī

[19] This section has been contributed by GIST Group. CDAC
[20] http://www.ethnologue.com/show_country.asp?name=in

Malayalam	mal	Kerala and Lakshadweep	Dravidian	Malayalam
Manipuri	mni	Meitei	Tibeto-Burman	Bangla Meitei-Meyek
Marathi	mar	Maharashtra	Indo-Aryan	Devanāgarī
Nepali	nep	Sikkim	Indo-Aryan	Devanāgarī
Oriya	ori	Orissa	Indo-Aryan	Oriya
Punjabi	pan	Punjab	Indo-Aryan	Gurmukhi
Sanskrit	san	Pan-Indian	Indo-Aryan	Devanāgarī
Santali	sat	Jharkhand	Munda	Ol Ciki
Sindhi	snd	Pan-Indian	Indo-Aryan	Perso-Arabic Devanāgarī Gujarati Roman (Latin)
Tamil	tam	Tamil Nadu and Pondicherry	Dravidian	Tamil
Telugu	tel	Andhra Pradesh	Dravidian	Telugu
Urdu	urd	Jammu and Kashmir	Indo-Aryan	Perso-Arabic

Although these 22 languages belong to 4 distinct language families: Indo-Aryan, Dravidian, Munda and Tibeto-Burman, insofar as the writing system is concerned, two major families can be identified:
-Languages whose writing system has evolved from Brāhmī: e.g.. Hindi, Bangla, Punjabi and all the Dravidian languages
- Languages whose writing system is Perso-Arabic in nature. These are only three in number: Kashmiri, Sindhi, and Urdu. Of these Sindhi and Kashmiri can be written also using a Brāhmī based writing system viz. Devanāgarī .
Smaller sub-sets of writing systems can be seen in the case of languages such as Meitei and Ol Ciki which have indigenous script systems.

APPENDIX III:

Comments on the white paper on Definitions and Questions circulated at the ICANN meet in Singapore in June 2011[21]

PDF UNDER DELIBERATION. WILL BE CIRCULATED SEPARATELY

[21] With inputs from Dr N. Ostler and Mr. Andrew Sullivan.

Appendix IV:
List of visually “look-alike” characters in Devanāgarī

Character 1	Character 2
उ U+0909	ऊ U+090A
ङ U+0919	ड U+0921
ज U+091C	ञ U+091E
ब U+092C	व U+0935
ऋ U+090B	ॠ U+0960
थ U+0925	य U+092F
प U+092A	ष U+0937
भ U+092D	म U+092E
इ	ई

U+0907	U+0908
ए U+090F	ऐ U+0910
ओ U+0913	औ U+0914
क U+0915	फ U+092b
ट U+091F	ठ U+0920
त U+0924	ल U+0932
र U+0930	ऱ U+0931
ल U+0932	ळ U+0933

Appendix V

Topics extraneous to the Variant Issues Project, but deemed to be of interest.

Issues which are extraneous to the Variant Issues report but in which variants are involved, are presented here.

1. REGISTRY MANAGEMENT

Registry Management of ABNF[22], Restriction rules, Language Tables and Variant Tables

The issues arising from delegation of Devanāgarī labels were discussed above. These are closely allied to the issues arising from the manner in which the language and variant tables will be managed by the registry. This discussion is limited to the policy for भारत, although the issues raised, because of their generic nature, can have larger ramifications.

Some of the major issues that arise are as under:

1. In the case of Devanāgarī, a large number of languages use the code block U+900. Given that the registry for .भारत will have to provide language-wise solutions how will the registry maintain the language table ?
2. Corollary to the above, will the registry support a variant table for each language ? The Hindi variant table has only two types of variants, whereas Marathi, Konkani and Nepali admit also the third type of variant table (cf. Section 2.2 supra)
3. In the case of TLD's other than.भारत, which rules will apply? It is suggested that in this case ICANN should deploy the rules and variant tables defined for each script/language

2. "Localization" of WHOIS

The term "Localization" has been used for WHOIS but the issues go far beyond. Two cases can be identified:

1. The label has no variant. In that case the major issue would be that of displaying the Information. Should the information be displayed in the language/script. Here language assumes priority. A Konkani speaker would not like information to be displayed in Hindi and vice-versa. Localization and language-wise information pertaining to WHOIS becomes a prime issue
2. Assuming that a given registrant is allocated variants (with/without payment of fees), this allocation raises the following issues:
3. In a scenario where a user checks one variant should all the other variants linked to that variant be displayed. This becomes especially

[22] Cf. footnote 12 supra. *ABNF is an acronym for Augmented Backus-Naur Formalism evolved to handle the Indic Akshar. Apart from rules governing Letters (L) it also handles Hyphen (H) and Digit (D)*

important in case ZWJ/ZWNJ are admitted, since on screen both variants will look alike

e.g. In the case of a label such as गड्डा : pit

गड्डा (without ZWNJ) गड्डा (with ZWNJ) give the same visual result

4. Corollary to the above should the WHOIS information be the same for a given label and its variant or should it be different ? The choice made will affect the registry functioning.

5. In a scenario where a variant is either deprecated or added at a later stage, how does the registry display such information. Will the registry have a systematic "re-indexing" and if so what will be the costs arising from it in terms of economics and logistics ?

6. The above case scenarios (1-3) are for variants which have been accepted. In the case of Type 2 variants where the variant is automatically blocked, should the registry display such variants also ?