Network Working Group	H. Alvestrand, Ed.	
Internet-Draft	Google	
Intended status: Standards Track	C. Karp, Ed.	
Expires: August 17, 2008	Swedish Museum of Natural History	
	February 14, 2008	

TOC

An updated IDNA criterion for right-to-left scripts
draft-alvestrand-idna-bidi-04

Status of this Memo

Abstract

The use of right-to-left scripts in internationalized domain names has presented several challenges. This memo discusses some problems with these scripts, and some shortcomings in the 2003 IDNA BIDI criterion. Based on this discussion, it proposes a new BIDI criterion for IDNA labels.

Table of Contents

---

1. Introduction and problem description

TOC

1.1. Purpose and applicability

TOC

This document's purpose is to establish a test that can be applied to Internationalized Domain Name (IDN) labels in Unicode form (U-labels) containing right-to-left characters.
When labels pass the test, they can be used with a minimal chance of these labels being displayed in a confusing way by a bidirectional display algorithm. In order to achieve this stability, it is also necessary that the test be applied to labels occuring before or after the label containing right-to-left characters, which prohibits some LDH-labels that are permitted in other contexts.

TOC

1.2. Background and history

The IDNA specification "Stringprep", [RFC3454] (Hoffman, P. and M. Blanchet, “Preparation of Internationalized Strings ("stringprep"),” December 2002.) makes the following statement in its section 6 on the bidi algorithm, :

> 3) If a string contains any RandALCat character, a RandALCat character MUST be the first character of the string, and a RandALCat character MUST be the last character of the string.

(A RandAlCat character is a character with unambiguously right-to-left directionality.)
The reasoning behind this prohibition was to ensure that every component of a displayed domain name has an unambiguously preferred direction. However, this makes certain words in languages written with right-to-left scripts invalid as IDN labels, and in at least one case means that all the words of an entire language are forbidden as IDN labels.
This will be illustrated below with examples taken from the Dhivehi and Yiddish languages, as written with the Thaana and Hebrew scripts, respectively.
In investigating this problem, it was realized that the RFC 3454 specification did not exactly specify what the requirement to be fulfilled was, and therefore, it was impossible to tell whether a simple relaxation of the rule would continue to fulfil the requirement. A further investigation led to the conclusion that for one reasonable set of requirements, IDNA2003's BIDI restriction did not fulfil the requirements. This document therefore proposes replacing the RFC 3454 BIDI requirement in its entirety.
While the document proposes completely new text, most reasonable labels that were allowed under the old criterion will also be allowed under the new criterion, so the operational impact of the rule change is limited.

1.3. Terminology

TOC

In this memo, we use "network order" to describe the sequence of characters as transmitted on the wire or stored in a file; the terms "first", "next" and "previous" are used to refer to the relationship of characters in network order.
We use "display order" to talk about the sequence of characters as imaged on a display medium; the terms "left" and "right" are used to refer to the relationship of characters in display order.
Most of the time, the examples use the abbreviations for the Unicode Bidi classes to denote the directionality of the characters; in some examples, the convention that uppercase characters are of class R or

AL, and lowercase characters are of class L is used - thus, the example string ABC.abc would consist of 3 right-to-left characters and 3 left-to-right characters.
The other terminology used to describe IDNA concepts is defined in [I-D.klensin-idnabis-issues] (Klensin, J., “Internationalizing Domain Names for Applications (IDNA): Issues, Explanation, and Rationale,” February 2008.)

---

2. Detailed examples

TOC

2.1. Dhivehi

TOC

Dhivehi, the official language of the Maldives, is written with the Thaana script. This displays some of the characteristics of Arabic script, including its directional properties, and the indication of vowels by the diacritical marking of consonantal base characters. This marking is obligatory, and both double vowels and syllable-final consonants are indicated by the marking of special unvoiced characters. Every Dhivehi word therefore ends with a combining mark.
The word for "computer", which is romanized as "konpeetaru", is written with the following sequence of Unicode code points:

> U+0786 THAANA LETTER KAAFU (AL)
>
> U+07AE THAANA OBOFILI (NSM)
>
> U+0782 THAANA LETTER NOONU (AL)
>
> U+07B0 THAANA SUKUN (NSM)
>
> U+0795 THAANA LETTER PAVIYANI (AL)
>
> U+07A9 THAANA LETTER EEBEEFILI (AL)
>
> U+0793 THAANA LETTER TAVIYANI (AL)
>
> U+07A6 THAANA ABAFILI (NSM)
>
> U+0783 THAANA LETTER RAA (AL)
>
> U+07AA THANAA UBIUFILI (NSM)

The directionality class of U+07AA in the Unicode database is NSM (non-spacing mark), which is not R or AL; a conformant implementation of the

IDNA2003 algorithm will say that "this is not in RandALCat", and refuse to encode the string.

2.2. Yiddish

TOC

Yiddish is one of several languages written with the Hebrew script (others include Hebrew and Ladino). This is basically a consonantal alphabet (also termed an "abjad") but Yiddish is written using an extended form that is fully vocalic. The vowels are indicated in several ways, of which one is by repurposing letters that are consonants in Hebrew. Other letters are used both as vowels and consonants, with combining marks, called "points", used to differentiate between them. Finally, some base characters can indicate several different vowels, which are also disambiguated by combining marks. Pointed characters can appear in word-final position and may therefore also be needed at the end of labels. This is not an invariable attribute of a Yiddish string and there is thus greater latitude here than there is with Dhivehi.

The organization now known as the "YIVO Institute for Jewish Research" developed orthographic rules for modern Standard Yiddish during the 1930s on the basis of work conducted in several venues since earlier in that century. These are given in, "The Standardized Yiddish Orthography: Rules of Yiddish Spelling, 6th ed., YIVO Institute for Jewish Research, New York, 1999, ISBN 0-914512-25-0", ("SYO") and are taken as normatively descriptive of modern Standard Yiddish in any context where that notion is deemed relevant. They have been applied exclusively in all Yiddish dictionaries published since their establishment, and are similarly dominant in academic and bibliographic regards.

It therefore appears appropriate for this repertoire also to be supported fully by IDNA. This presents no difficulty with characters in initial and medial positions, but pointed characters are regularly used in final position as well. All of the characters in the SYO repertoire appear in both marked and unmarked form with one exception: the HEBREW LETTER PE (U+05E4). The SYO only permits this with a HEBREW POINT DAGESH (U+05BC), providing the Yiddish equivalent to the Latin letter "p", or a HEBREW POINT RAFE (U+05BF), equivalent to the Latin letter "f". There is, however, a separate unpointed allograph, the HEBREW LETTER FINAL PE (U+05E3), for the latter character when it appears in final position. The constraint on the use of the SYO repertoire resulting from the proscription of combining marks at the end of RTL strings thus reduces to nothing more, or less, than the equivalent of saying that a string of Latin characters cannot end with the letter "p". It must also be noted that the HEBREW LETTER PE with HEBREW POINT DAGESH is characteristic of almost all traditional Yiddish

orthographies that predate (or remain in use in parallel to) the SYO, being the first pointed character to appear in any of them.
A more general instantiation of the basic problem can be seen in the representation of the YIVO acronym. This is written with the Hebrew letters YOD YOD HIRIQ VAV VAV ALEF QAMATS, where HIRIQ and QAMATS are combining points:

> U+05D9 HEBREW LETTER YOD (R)
>
> U+05B4 HEBREW POINT HIRIQ (NSM)
>
> U+05D5 HEBREW LETTER VAV (R)
>
> U+05D0 HEBREW LETTER ALEF (R)
>
> U+05B8 HEBREW POINT QAMATS (NSM)

The directionality class of U+05B8 HEBREW POINT QAMATS in the Unicode database is NSM, which again causes the IDNA2003 algorithm to reject the string.
It may also be noted that all of the combined characters mentioned above exist in precomposed form at separate positions in the Unicode chart. However, by invoking Stringprep, the IDNA2003 algorithm also rejects those codepoints, for reasons not discussed here.

2.3. Strings with numbers

TOC

RFC 3454, in its insistence that the first or last character of a string be category R or AL, prohibited strings that contained right-to-left characters and numbers at the end.
Consider the strings ALEF 5 (HEBREW LETTER ALEF + DIGIT FIVE) and 5 ALEF. Displayed in a LTR context, the first one will be displayed from left to right as 5 ALEF (with the 5 being considered right-to-left because of the leading ALEF), while 5 ALEF will be displayed in exactly the same order (5 taking the direction from context). Clearly, only one of those should be permitted as a registered label.

---

3. An expanded justification for the bidi rule

TOC

One issue with RFC 3454 was that it did not give an explicit justification for the bidi rule, thus it was hard to tell if a modified rule would continue to fulfil the purpose for which the RFC 3454 rule was written.

This document proposes an explicit justification, by stating a set of requirements for which it is possible to test whether or not the modified rule fulfils the requirement.
All the text in this document assumes that text containing the labels under consideration will be displayed using the Unicode bidirectional algorithm [UAX9] (Davis, M., “Unicode Standard Annex #9: The Bidirectional Algorithm, revision 15,” 03 2005.).
The justification proposed is this:

- No two labels, when presented in display order, should have the same sequence of characters without also having the same sequence of characters in network order. (This is the criterion that is explicit in RFC 3454).

- In a display of a string of labels, the characters of each label should remain grouped between the characters delimiting the labels.

- These properties should hold true both when the string is embedded in a paragraph with LTR direction and when it's embedded in a paragraph with RTL direction, as long as explicit directional controls are not used within the same paragraph.

Several stronger statements were considered and rejected, because they seem to be impossible to fulfil within the constraints of the Unicode bidirectional algorithm. These include:

- The appearance of a label should be unaffected by its embedding context. This proved impossible even for ASCII labels; the label "123-456" will have a different display order in an RTL context than in a LTR context.

- The sequence of labels should be consistent with network order. This proved impossible - a domain name consisting of the labels (in network order) L1.R1.R2.L2 will be displayed as L1.R2.R1.L2 in an LTR context.

- The "remain grouped" property should remain true when directional controls (LRE, RLE, RLO, LRO, PDF) are used in the same paragraph (outside of the labels). Because these controls affect presentation order in non-obvious ways, by affecting the "sor" and "eor" properties of the Unicode BIDI algorithm, the conditions above would be very hard to satisfy for an useful set of strings if this was true. As long as these controls have no influence over the display of the domain name, no problem will be caused, but the exact criterion for "will not influence" is hard to codify.

- The "no two labels display the same" should hold true between LTR paragraphs and RTL paragraphs. This was shown to be unsound.

*No two domain names should be displayed the same, even under differing directionality. This was shown to be unsound, since the domain name (network) ABC.abc will have display order CBA.abc in an LTR context and abc.CBA in an RTL context, while the domain name (network) abc.ABC will display as abc.CBA in an LTR context and as CBA.abc in an RTL context.

For reference, here are the values that the Unicode BIDI property can have:

*L - Left-to-right - most letters in LTR scripts

*R - Right-to-left - most letters in non-Arabic RTL scripts

*AL - Arabic letters - most letters in the Arabic script

*EN - European Number (0-9)

*ES - European Number Separator (+ and -)

*ET - European Number Terminator (currency symbols, the hash sign, the percent sign and so on)

*AN - Arabic Number

*CS - Common Number Separator (. , / : et al)

*NSM - Nonspacing Mark - most combining accents

*BN - Boundary Neutral - control characters

*B - Paragraph Separator

*S - Segment Separator

*WS - Whitespace, including the SPACE character

*ON - Other Neutrals, including @, &, parentheses, MIDDLE DOT

*LRE, LRO, RLE, RLO, PDF - these are "directional control characters", and are not used in IDNA labels.

The "remain grouped" property can be more formally stated as:

- *Let "Delimiter chars" be a set of characters with the Unicode BIDI properties CS, WS, ON. (These are commonly used to delimit labels - both the FULL STOP and the space are included.)
  - -ET, though it commonly occurs next to domain names in practice, is problematic: the context R CS L EN ET (for instance A.a1%) makes the label L EN grow unstable.
  - -ES commonly occurs in labels as HYPHEN-MINUS, but could also be used as a delimiter (for instance, the plus sign). It is left out here.
- *Let "Position" be the position of a character in a string (in network order)
- *Let "Bidi position" be the position computed by the Unicode Bidi algorithm

In a paragraph with an embedded string formed from the substrings A B L C D, where A and D are (possibly zero-length) legal labels, and B and C are single "Delimiter chars", the label L is a legal label if, for all A, B, C and D, the bidi position of all characters in L is within the range of positions for the characters of L in the string, for both the LTR and RTL paragraph direction.
(The "zero-length" case represents the case where a domain name is next to something that isn't a domain name, separated by a delimiter character).
The "No two labels" property can be formally stated as:
If two labels L and L', embedded as for the test above, displayed in a paragraph with the same directionality, are rearranged into the same sequence of codepoints, neither L nor L' is a legal label.

---

4. A replacement for the RFC 3454 criterion

TOC

A set of rules that satisfies the tests above is as follows. The main bullets give the rule, subordinate bullets (if any) give justifications or examples of things that break if this rule is not present. The term "unstable" means that it fails to satisfy the "remain grouped" property defined above.
Exhaustive testing has verified that strings that satisfy this criterion satisfy both the requirements above at least for all strings up to 6 characters.

*Only characters with the BIDI properties L, R, AL, AN, EN, ES, BN, ON and NSM are allowed.

  -B, S and WS are excluded because they are separators or spaces.

  -LRE, LRO, RLE, RLO, PDF are excluded because they are bidi controls.

  -ET is excluded because the string L ET is unstable.

  -CS is excluded because the string L CS is unstable.

*ES and ON are not allowed in the first position

  -ES R and ON R are both unstable.

*ES and ON, followed by zero or more NSM, is not allowed in the last position

  -L ON and L ES are both unstable.

*If an L is present, no R, AL or AN may be present, and vice versa.

*If an EN is present, no AN may be present, and vice versa.

*The first character may not be an NSM

*The first character may not be an EN (European Number) or an AN (Arabic Number).

  -If the character on both sides of a CS is an EN or an AN, the labels turn unstable.

  -Some domain names where some of the labels use leading EN and AN may be problem-free, but there's no way of verifying this while looking at a single label in isolation.

  -NOTE: This is a restriction on ASCII labels when used together with IDNA labels. This is a change from the existing rules for ASCII labels.

  -We could achieve stability by barring numbers at the end of labels, but this may be more disruptive in practice.

TOC

5. Other issues in need of resolution

This document concerns itself only with the rules that are needed when dealing with domain names with characters that have differing Bidi properties, and considers characters only in terms of their Bidi properties. All other issues with these scripts have to be considered in other contexts.
Another set of issues concerns the proper display of IDNs with a mixture of LTR and RTL labels, or only RTL labels.
It is unrealistic to expect that domain names will be written using embedded formatting codes between their labels; thus, the display order will be determined by the bidirectional algorithm. Thus, a sequence (in network order) of R1.R2.ltr will be displayed in the order 2R.1R.ltr in a LTR context, which might surprise someone expecting to see labels displayed in hierarchical order. Again, this memo does not attempt to suggest a solution to this problem.

---

6. Compatibility considerations

TOC

6.1. Backwards compatibility considerations

TOC

As with any change to an existing standard, it is important to consider what happens with existing implementations when the change is introduced. The following troublesome cases have been noted:

- Old program used to input the newly allowed string. If the old program checks the input against RFC 3454, the string will not be allowed, and that domain name will remain inaccessible.

- Old program is asked to display the newly allowed string, and checks it against RFC 3454 before displaying. The program will perform some kind of fallback, most likely displaying the Punycode form of the string.

- Old program tries to display the newly allowed string. If the old program has code for displaying the last character of a string that is different from the code used to display the characters in the middle of the string, display may be inconsistent and cause confusion.

One particular example of the last case is if a program chooses to examine the last character (in network order) of a string in order to determine its directionality, rather than its first; if it finds an NSM

character and tries to display the string as if it was a left-to-right string, the resulting display may be interesting, but not useful.
The editors believe that these cases will have less harmful impact in practice than continuing to deny the use of words from the languages for which these strings are necessary as IDN labels.
This specification forbids using leading European numbers in ASCII-only labels; this is in conflict with a large installed base of such labels. The harm resulting from violating this rule is seen when a label at the next level down in the hierarchy ends with a number (Arabic or European). Zone managers, both registries and private zone managers, can check for this particular condition before they allow registration of any string with right-to-left characters in it; generally it is best to not allow registration of any right-to-left strings in a zone where the label at the level above begins with a digit.

6.2. Forward compatibiltiy considerations

TOC

This text is, intentionally, specified strictly in terms of the Unicode BIDI properties. The determination that the condition is sufficient to fulfil the criteria depends on the Unicode BIDI algorithm; it is unlikely that drastic changes will be made to this algorithm.
However, the determination of validity for any string depends on the Unicode BIDI property values, which are not declared immutable by the Unicode Consortium. Furthermore, the behaviour of the algorithm for any given character is likely to be linguistically and culturally sensitive, so that it's not unlikely that later versions of the Unicode standard may change the bidi properties assigned to certain Unicode characters.
This memo does not propose a solution for this problem.

---

7. IANA Considerations

TOC

This document makes no request of IANA.
Note to RFC Editor: this section may be removed on publication as an RFC.

---

8. Security Considerations

TOC

This modification will allow some strings to be used in Stringprep contexts that are not allowed today. It is possible that differences in the interpretation of the specification between old and new

implementations could pose a security risk, but it is difficult to envision any specific instantiation of this.
Any rational attempt to compute, for instance, a hash over an identifier processed by Stringprep would use network order for its computation, and thus be unaffected by the changes proposed here.
While it is not believed to pose a problem, if display routines had been written with specific knowledge of the RFC 3454 Stringprep prohibitions, it is possible that the potential problems noted under "backwards compatibility" could cause new kinds of confusion.
The rule about leading numbers, which is more restrictive than current practice for domain names, has a peculiar interaction with the DNAME record; a DNAME record can point to a zone where right-to-left labels are registered without the knowledge or consent of the zone owner; if the name of the DNAME begins with a number, this can cause display of the right-to-left labels in the zone to be confusing. It is recommended that DNAMEs pointing to zones allowing right-to-left labels should not start with a digit, but a pointed-to zone owner has no way of enforcing this.

---

9. Acknowledgements

TOC

While the listed editors held the pen, this document represents the joint work and conclusions of an ad hoc design team. In addition to the editors this consisted of, in alphabetic order, Tina Dam, Patrik Faltstrom, and John Klensin. Many further specific contributions and helpful comments were received from the people listed below, and others who have contributed to the development and use of the IDNA protocols. The team wishes in particular to thank Roozbeh Pournader for calling its attention to the issue with the Thaana script, Paul Hoffmann for pointing out the need to be explicit about backwards compatibility considerations, Ken Whistler for suggesting the basis of the formalized "remain grouped" requirement, and Erik van der Poel for careful review, comments and verification of the rulesets.

Appendix A. Change log

TOC

This appendix is intended to be removed when this document is published as an RFC.

TOC

A.1. Changes from -00 to -01

Suggested a possible new algorithm.
Multiple smaller changes.

A.2. Changes from -01 to -02

TOC

Date of publication updated.
Change log added.

A.3. Changes from -02 to -03

TOC

Intro changed to reflect addressing the deeper issues with the Bidi algorithm.
Gave formalized criteria for "valid strings", and documented the new set of requirements for strings that satisfy the criteria.
Removed most of section 5, "Other problems", and noted that this memo focuses ONLY on issues that can be evaluated by looking at the bidi properties of characters.

A.4. Changes from -03 to -04

TOC

Added back AN to the list of allowed characters; it had been left out by accident in -03.
Removed some rules that were redundant.
Added some considerations for backwards compatibility and interaction with ASCII labels that start with a number.
Mentioned the issue with DNAME pointing to a zone containing RTL labels in the security considerations section.
Wording updates in multiple places, including some spelling errors.
Rewrote the introduction section.
Split references into "normative" and "informative".

---

10. References

TOC

10.1. Normative references

TOC

[I-D.klensin-idnabis-issues]	Klensin, J., “Internationalizing Domain Names for Applications (IDNA): Issues, Explanation, and Rationale,” draft-klensin-idnabis-issues-07 (work in progress), February 2008 (TXT).
[UAX9]	Davis, M., “Unicode Standard Annex #9: The Bidirectional Algorithm, revision 15,” 03 2005.

10.2. Informative references

TOC

[RFC3454]	Hoffman, P. and M. Blanchet, “Preparation of Internationalized Strings ("stringprep"),” RFC 3454, December 2002 (TXT).

Authors' Addresses

TOC

	Harald Tveit Alvestrand (editor)
	Google
	Beddingen 10
	Trondheim, 7014
	Norway
Email:	harald@alvestrand.no
	Cary Karp (editor)
	Swedish Museum of Natural History
	Frescativ. 40
	Stockholm, 10405
	Sweden
Phone:	+46 8 5195 4055
Fax:	
Email:	ck@nrm.museum
URI:	

Full Copyright Statement

TOC

OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.