Internet Engineering Task Force (IETF)
Request for Comments: 5893
Category: Standards Track
ISSN: 2070-1721

H. Alvestrand, Ed.
Google
C. Karp
Swedish Museum of Natural History

August 2010

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

Abstract

The use of right-to-left scripts in Internationalized Domain Names (IDNs) has presented several challenges. This memo provides a new Bidi rule for Internationalized Domain Names for Applications (IDNA) labels, based on the encountered problems with some scripts and some shortcomings in the 2003 IDNA Bidi criterion.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5893.

Table of Contents

1. Introduction

1.1. Purpose and Applicability

The purpose of this document is to establish a rule that can be applied to Internationalized Domain Name (IDN) labels in Unicode form (U-labels) containing characters from scripts that are written from right to left. It is part of the revised IDNA protocol [RFC5891].

When labels satisfy the rule, and when certain other conditions are satisfied, there is only a minimal chance of these labels being displayed in a confusing way by the Unicode bidirectional display algorithm.

The other normative documents in the IDNA2008 document set establish criteria for valid labels, including listing the permitted characters. This document establishes additional validity criteria for labels in scripts normally written from right to left.

This specification is not intended to place any requirements on domain names that do not contain characters from such scripts.

1.2. Background and History

The "Stringprep" specification [RFC3454], part of IDNA2003, made 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 made certain words in languages written with right-to-left scripts invalid as IDN labels, and in at least one case (Dhivehi) meant that all the words of an entire language were forbidden as IDN labels.

This is illustrated below with examples taken from the Dhivehi and Yiddish languages, as written with the Thaana and Hebrew scripts, respectively.

RFC 3454 did not explicitly state the requirement to be fulfilled. Therefore, it is impossible to determine whether a simple relaxation of the rule would continue to fulfill the requirement.

While this document specifies rules quite different from RFC 3454, most reasonable labels that were allowed under RFC 3454 will also be allowed under this specification (the most important example of non-permitted labels being labels that mix Arabic and European digits (AN and EN) inside an RTL label, and labels that use AN in an LTR label -- see Section 1.4 for terminology), so the operational impact of using the new rule in the updated IDNA specification is limited.

1.3. Structure of the Rest of This Document

Section 2 defines a rule, the "Bidi rule", which can be used on a domain name label to check how safe it is to use in a domain name of possibly mixed directionality. The primary initial use of this rule is as part of the IDNA2008 protocol [RFC5891].

Section 3 sets out the requirements for defining the Bidi rule.

Section 4 gives detailed examples that serve as justification for the new rule.

Section 5 to Section 8 describe various situations that can occur when dealing with domain names with characters of different directionality.

Only Section 1.4 and Section 2 are normative.

1.4. Terminology

The terminology used to describe IDNA concepts is defined in the Definitions document [RFC5890].

The terminology used for the Bidi properties of Unicode characters is taken from the Unicode Standard [Unicode52].

The Unicode Standard specifies a Bidi property for each character. That property controls the character's behavior in the Unicode bidirectional algorithm [Unicode-UAX9]. 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, and Extended Arabic-Indic numbers)
- ES - European Number Separator (+ and -)
- ET - European Number Terminator (currency symbols, the hash sign, the percent sign and so on)
- AN - Arabic Number; this encompasses the Arabic-Indic numbers, but not the Extended Arabic-Indic numbers
- CS - Common Number Separator (. , / : et al)
- NSM - Nonspacing Mark - most combining accents
- BN - Boundary Neutral - control characters (ZWNJ, ZWJ, and others)
- 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.

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", "previous", "beginning", "end", "before", and "after" are used to refer to the relationship of characters and labels 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 and labels in display order.

Most of the time, the examples use the abbreviations for the Unicode Bidi classes to denote the directionality of the characters; the example string CS L consists of one character of class CS and one character of class L. 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 three right-to-left characters and three left-to-right characters.

The directionality of such examples is determined by context -- for instance, in the sentence "ABC.abc is displayed as CBA.abc", the first example string is in network order, the second example string is in display order.

The term "paragraph" is used in the sense of the Unicode Bidi specification [Unicode-UAX9]. It means "a block of text that has an overall direction, either left to right or right to left", approximately; see the "Unicode Bidirectional Algorithm" [Unicode-UAX9] for details.

"RTL" and "LTR" are abbreviations for "right to left" and "left to right", respectively.

An RTL label is a label that contains at least one character of type R, AL, or AN.

An LTR label is any label that is not an RTL label.

A "Bidi domain name" is a domain name that contains at least one RTL label. (Note: This definition includes domain names containing only dots and right-to-left characters. Providing a separate category of "RTL domain names" would not make this specification simpler, so it has not been done.)

2. The Bidi Rule

The following rule, consisting of six conditions, applies to labels in Bidi domain names. The requirements that this rule satisfies are described in Section 3. All of the conditions must be satisfied for the rule to be satisfied.

1. The first character must be a character with Bidi property L, R, or AL. If it has the R or AL property, it is an RTL label; if it has the L property, it is an LTR label.

2. In an RTL label, only characters with the Bidi properties R, AL, AN, EN, ES, CS, ET, ON, BN, or NSM are allowed.

3. In an RTL label, the end of the label must be a character with Bidi property R, AL, EN, or AN, followed by zero or more characters with Bidi property NSM.

4. In an RTL label, if an EN is present, no AN may be present, and vice versa.

5. In an LTR label, only characters with the Bidi properties L, EN, ES, CS, ET, ON, BN, or NSM are allowed.

6. In an LTR label, the end of the label must be a character with Bidi property L or EN, followed by zero or more characters with Bidi property NSM.

The following guarantees can be made based on the above:

- In a domain name consisting of only labels that satisfy the rule, the requirements of Section 3 are satisfied. Note that even LTR labels and pure ASCII labels have to be tested.

- In a domain name consisting of only LDH labels (as defined in the Definitions document [RFC5890]) and labels that satisfy the rule, the requirements of Section 3 are satisfied as long as a label that starts with an ASCII digit does not come after a right-to-left label.

No guarantee is given for other combinations.

3. The Requirement Set for the Bidi Rule

This document, unlike RFC 3454 [RFC3454], provides an explicit justification for the Bidi rule, and states a set of requirements for which it is possible to test whether or not the modified rule fulfills the requirement.

All the text in this document assumes that text containing the labels under consideration will be displayed using the Unicode bidirectional algorithm [Unicode-UAX9].

The requirements proposed are these:

- Label Uniqueness: No two labels, when presented in display order in the same paragraph, should have the same sequence of characters without also having the same sequence of characters in network order, both when the paragraph has LTR direction and when the paragraph has RTL direction. (This is the criterion that is explicit in RFC 3454). (Note that a label displayed in an RTL paragraph may display the same as a different label displayed in an LTR paragraph and still satisfy this criterion.)

- Character Grouping: When displaying a string of labels, using the Unicode Bidi algorithm to reorder the characters for display, the characters of each label should remain grouped between the characters delimiting the labels, both when the string is embedded in a paragraph with LTR direction and when it is embedded in a paragraph with RTL direction.

Several stronger statements were considered and rejected, because they seem to be impossible to fulfill 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-A" will have a different display order in an RTL context than in an LTR context. (This particular example is, however, disallowed anyway.)

- The sequence of labels should be consistent with network order. This proved impossible -- a domain name consisting of the labels (in network order) L1.R2.R3.L4 will be displayed as L1.R3.R2.L4 in an LTR context. (In an RTL context, it will be displayed as L4.R3.R2.L1).

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

One possible requirement was thought to be problematic, but turned out to be satisfied by a string that obeys the proposed rules:

- The Character Grouping requirement should be satisfied 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 require extra testing in order to figure out whether or not they influence the display of the domain name. Testing found that for the strings allowed under the rule presented in this document, directional controls do not influence the display of the domain name.

This is still not stated as a requirement, since it did not seem as important as the stated requirements, but it is useful to know that Bidi domain names where the labels satisfy the rule have this property.

In the following descriptions, first-level bullets are used to indicate rules or normative statements; second-level bullets are commentary.

The Character Grouping requirement can be more formally stated as:

- Let "Delimiterchars" 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. They are not allowed in domain labels.)

  * 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 not satisfy the character grouping requirement.

  * 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 "unproblematic label" be a label that either satisfies the requirements or does not contain any character with the Bidi properties R, AL, or AN and does not begin with a character with the Bidi property EN. (Informally, "it does not start with a number".)

A label X satisfies the Character Grouping requirement when, for any Delimiter Character D1 and D2, and for any label S1 and S2 that is an unproblematic label or an empty string, the following holds true:

If the string formed by concatenating S1, D1, X, D2, and S2 is reordered according to the Bidi algorithm, then all the characters of X in the reordered string are between D1 and D2, and no other characters are between D1 and D2, both if the overall paragraph direction is LTR and if the overall paragraph direction is RTL.

Note that the definition is self-referential, since S1 and S2 are constrained to be "legal" by this definition. This makes testing changes to proposed rules a little complex, but does not create problems for testing whether or not a given proposed rule satisfies the criterion.

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.

Note about the position of BN: The Unicode bidirectional algorithm specifies that a BN has an effect on the adjoining characters in network order, not in display order, and are therefore treated as if removed during Bidi processing ([Unicode-UAX9], Section 3.3.2, rule X9 and Section 5.3). Therefore, the question of "what position does a BN have after reordering" is not meaningful. It has been ignored while developing the rules here.

The Label Uniqueness requirement can be formally stated as:

If two non-identical labels X and Y, embedded as for the test above, displayed in paragraphs with the same directionality, are reordered by the Bidi algorithm into the same sequence of code points, the labels X and Y cannot both be legal.

4. Examples of Issues Found with RFC 3454

4.1. Dhivehi

Dhivehi, the official language of the Maldives, is written with the Thaana script. This script displays some of the characteristics of the 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 two consecutive vowels and syllable-final consonants are indicated with unvoiced combining marks. 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 THAANA UBUFILI (NSM)

The directionality class of U+07AA in the Unicode database [Unicode52] is NSM (Nonspacing 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.

4.2. Yiddish

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, one of which 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" [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 formal 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 the 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 acronym is written with the Hebrew letters YOD YOD HIRIQ VAV VAV ALEF QAMATS, where HIRIQ and QAMATS are combining points. The Unicode code points are:

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 code points, for reasons not discussed here.

4.3. Strings with Numbers

By requiring that the first or last character of a string be a member of category R or AL, the Stringprep specification [RFC3454] prohibited a string containing right-to-left characters from ending with a number.

Consider the strings ALEF 5 (HEBREW LETTER ALEF + DIGIT FIVE) and 5 ALEF. Displayed in an 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, but barring them both seems unnecessary.

5. Troublesome Situations and Guidelines

There are situations in which labels that satisfy the rule above will be displayed in a surprising fashion. The most important of these is the case where a label ending in a character with Bidi property AL, AN, or R occurs before a label beginning with a character of Bidi property EN. In that case, the number will appear to move into the label containing the right-to-left character, violating the Character Grouping requirement.

If the label that occurs after the right-to-left label itself satisfies the Bidi criterion, the requirements will be satisfied in all cases (this is the reason why the criterion talks about strings containing L in some cases). However, the IDNABIS WG concluded that this could not be required for several reasons:

- There is a large current deployment of ASCII domain names starting with digits. These cannot possibly be invalidated.

- Domain names are often constructed piecemeal, for instance, by combining a string with the content of a search list. This may occur after IDNA processing, and thus in part of the code that is not IDNA-aware, making detection of the undesirable combination impossible.

- Even if a label is registered under a "safe" label, there may be a DNAME [RFC2672] with an "unsafe" label that points to the "safe" label, thus creating seemingly valid names that would not satisfy the criterion.

- Wildcards create the odd situation where a label is "valid" (can be looked up successfully) without the zone owner knowing that this label exists. So an owner of a zone whose name starts with a digit and contains a wildcard has no way of controlling whether or not names with RTL labels in them are looked up in his zone.

Rather than trying to suggest rules that disallow all such undesirable situations, this document merely warns about the possibility, and leaves it to application developers to take whatever measures they deem appropriate to avoid problematic situations.

6. 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 scripts that are written from right to left must be considered in other contexts.

One such issue is the need to keep numbers separate. Several scripts are used with multiple sets of numbers -- most commonly they use Latin numbers and a script-specific set of numbers, but in the case of Arabic, there are two sets of "Arabic-Indic" digits involved.

The algorithm in this document disallows occurrences of AN-class characters ("Arabic-Indic digits", U+0660 to U+0669) together with EN-class characters (which includes "European" digits, U+0030 to U+0039 and "extended Arabic-Indic digits", U+06F0 to U+06F9), but does not help in preventing the mixing of, for instance, Bengali digits (U+09E6 to U+09EF) and Gujarati digits (U+0AE6 to U+0AEF), both of which have Bidi class L. A registry or script community that wishes to create rules restricting the mixing of digits in a label will be able to specify these restrictions at the registry level. Some rules are also specified at the protocol level.

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 applications will display domain names using embedded formatting codes between their labels (for one thing, no reliable algorithms for identifying domain names in running text exist); thus, the display order will be determined by the Bidi algorithm. Thus, a sequence (in network order) of R1.R2.ltr will be

displayed in the order 2R.1R.ltr in an LTR context, which might surprise someone expecting to see labels displayed in hierarchical order. People used to working with text that mixes LTR and RTL strings might not be so surprised by this. Again, this memo does not attempt to suggest a solution to this problem.

7. Compatibility Considerations

7.1. Backwards Compatibility Considerations

As with any change to an existing standard, it is important to consider what happens with existing implementations when the change is introduced. Some troublesome cases include:

- An old program used to input the newly allowed label. If the old program checks the input against RFC 3454, some labels will not be allowed, and domain names containing those labels will remain inaccessible.

- An old program is asked to display the newly allowed label, and checks it against RFC 3454 before displaying. The program will perform some kind of fallback, most likely displaying the label in A-label form.

- An old program tries to display the newly allowed label. If the old program has code for displaying the last character of a label that is different from the code used to display the characters in the middle of the label, the 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 a 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 does not forbid using leading European digits in ASCII-only labels, since this would conflict with a large installed base of such labels, and would increase the scope of the specification from RTL labels to all labels. The harm resulting from this limitation of scope is described in Section 5. Registries and private zone managers can check for this particular condition before they allow registration of any RTL label. Generally, it is best to

disallow registration of any right-to-left strings in a zone where the label at the level above begins with a digit.

7.2. Forward Compatibility Considerations

This text is intentionally specified strictly in terms of the Unicode Bidi properties. The determination that the condition is sufficient to fulfill 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 behavior of the algorithm for any given character is likely to be linguistically and culturally sensitive, so while it should occur rarely, it is possible 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.

8. Security Considerations

The display behavior of mixed-direction text can be extremely surprising to users who are not used to it; for instance, cut and paste of a piece of text can cause the text to display differently at the destination, if the destination is in another directionality context, and adding a character in one place of a text can cause characters some distance from the point of insertion to change their display position. This is, however, not a phenomenon unique to the display of domain names.

The new IDNA protocol, and particularly these new Bidi rules, will allow some strings to be used in IDNA contexts that are not allowed today. It is possible that differences in the interpretation of labels between implementations of IDNA2003 and IDNA2008 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 IDNA would use network order for its computation, and thus be unaffected by the new rules proposed here.

While it is not believed to pose a problem, if display routines had been written with specific knowledge of the RFC 3454 IDNA prohibitions, it is possible that the potential problems noted under "Backwards Compatibility Considerations" could cause new kinds of confusion.

9. Acknowledgements

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 particular formulation of the Bidi rule in Section 2 was suggested by Matitiahu Allouche.

The team wishes, in particular, to thank Roozbeh Pournader for calling its attention to the issue with the Thaana script, Paul Hoffman for pointing out the need to be explicit about backwards compatibility considerations, Ken Whistler for suggesting the basis of the formalized "Character Grouping" requirement, Mark Davis for commentary, Erik van der Poel for careful review, comments, and verification of the rulesets, Marcos Sanz, Andrew Sullivan, and Pete Resnick for reviews, and Vint Cerf for chairing the working group and contributing massively to getting the documents finished.

10. References

10.1. Normative References

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[Unicode-UAX9] The Unicode Consortium, "Unicode Standard Annex #9: Unicode Bidirectional Algorithm", September 2009, <http://www.unicode.org/reports/tr9/>.

[Unicode52] The Unicode Consortium. The Unicode Standard, Version 5.2.0, defined by: "The Unicode Standard, Version 5.2.0", (Mountain View, CA: The Unicode Consortium, 2009. ISBN 978-1-936213-00-9). <http://www.unicode.org/versions/Unicode5.2.0/>.

10.2. Informative References

[RFC2672] Crawford, M., "Non-Terminal DNS Name Redirection", RFC 2672, August 1999.

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

[RFC5891] Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol", RFC 5891, August 2010.

[SYO] "The Standardized Yiddish Orthography: Rules of Yiddish Spelling, 6th ed., New York, ISBN 0-914512-25-0", 1999.

Authors' Addresses

Harald Tveit Alvestrand (editor)
Google
Beddingen 10
Trondheim, 7014
Norway

EMail: harald@alvestrand.no

Cary Karp
Swedish Museum of Natural History
Frescativ. 40
Stockholm, 10405
Sweden

Phone: +46 8 5195 4055
Fax:
EMail: ck@nic.museum