The Unicode Consortium Discussion Forum

Forum Home **Unicode Home Page** **Code Charts** **Technical Reports** **FAQ Pages**

Forum FAQ - **Smartfeed** - **0 new messages** - **Search** - **Members** - **User Control Panel** - **Logout [Sarasvati]**

Last visit was: Mon Oct 31, 2011 8:36 pm It is currently Tue Nov 01, 2011 9:47 am

View unanswered posts | View active topics View new posts | View your posts

Board index » Public Review Discussions » PRI 185 - Extension of UBA for improved display of URL/IRIs

All times are UTC - 8 hours

Forum rules

Please click here to view the forum rules

October 2011 feedback on PRI 185 (long!)

[Moderator Control Panel]

new topic post reply **Page 1 of 1** [2 posts]

Subscribe topic | Bookmark topic | Print view | E-mail friend **Previous topic | Next topic**

Author **Message**

MartinJD

offline

Joined: Tue Dec 07, 2010 10:49 pm
Posts: 3
Location: Japan

Post subject: October 2011 feedback on PRI 185 (long!)
Posted: Mon Oct 24, 2011 1:26 am

These are my comments on "Extension of UBA for improved display of URL/IRIs", available from http://www.unicode.org/review/pri185/, as modified on Sept. 22 (presumably 2011).

I have commented mostly on procedural issues in the last round, but have taken a deeper look at technical and editorial issues this time, too. I wrote part of this on a very long flight, so some references are missing; if you need some additional pointers, don't hesitate to ask.

Procedural Issues
================

Opening the ability to comment via the Unicode Forum is some progress on the previous way of commenting via a Web form (which was essentially a black hole for outsiders). However, it is still very much a one-way street.

This makes it difficult to involve affected communities such as the IETF

Working Groups (WGs, or former WG) on Internationalized Domain Names, Email Address Internationalization, and Internationalized Resource Identifiers, the relevant groups at W3C, and in many ways most important, the actually affected users that use bidi IRIs.

It also makes it difficult to find out how, and more importantly, why, comments have been addressed or not. It is still a far way away from how other organizations deal with public comments. In the W3C, providing a public list of all public comments and how they are addressed is standard practice. In the IETF, most of the discussions are held on public mailing list, and many WGs are using a public tracker (http://trac.tools.ietf.org/).

Preexisting Specs and Parallel Work
==================================

The IRI specification (RFC 3987, http://tools.ietf.org/html/rfc3987) as well as draft updates (http://tools.ietf.org/html/draft-ietf-iri-3987bis), and the specifications for Email Address Internationalization as well as their draft updates should be referenced at the start of the document. (There are no specs for file names as far as I know.) It's a good idea to point readers to more introductory material such as Richard Ishida's "idn-and-iri", but that's not enough for what is supposed to become (part of) a spec itself.

RFC 3987 contains a section about the display of bidirectional IRIs (Section 4, http://tools.ietf.org/html/rfc3987#section-4.4). This should clearly be mentioned in the document. The section was written based on the following goals/assumptions:

1) That it would be desirable that bidi IRIs were displayed the same everywhere, both in places where they are identified as such (e.g. a browser's address/location bar) and in free text where no special processing could be applied to them.

2) That it was unfeasible to change the Unicode Bidirectional Algorithm (UBA) to deal with IRIs as a special case.

The first assumption is shared by the current proposal; removing the second assumption is at the base of the current proposal.

Now that changing (or extending) the UBA is on the table, we have to check what needs specifying, and where. My current take is that we have the following pieces:

1) Display of bidi IRIs once identified: UBA extension, with strong input from stakeholders in affected regions and from IRI WG.

2) Identifying IRIs in contexts: This would ideally be provided by the IETF. There is Appendix C of the URI spec (http://tools.ietf.org/html/rfc3986#appendix-C), Delimiting a URI in Context, and there was at least one attempt to do something in this direction (see http://tools.ietf.org/html/draft-yoneya ... gnition-00), but no wider interest and no pressure for standardization (the functionality seemed to work well where needed (e.g. email programs) and minor differences in implementation seemed to hurt nobody). So there's a rather large chance that this remains for the UTC to do, although with strong input from the IETF.

3) Restrictions on strong directionality mixing for components such as domain name labels: This is done for IDNA in RFC 5893 (http://tools.ietf.org/html/rfc5893) and is being updated and adapted based on the RFC 5893 effort for IRIs in the IRI WG. Input from the bidi experts in the UTC is greatly appreciated.

We should make sure that we have got something like the above "pieces of the puzzle" right before we get too much into specific technical details.

Document Target
===============

There is talk about this being an experimental extension. Care should be given to be extremely clear what these two words mean, in particular because I don't know any other cases where this has been done in an Unicode context.

Extension seems to mean "the bidi algorithm can be used with or without this". This is desirable from an implementer's perspective, but not from a security perspective.

Experimental seems to mean "we aren't really sure yet whether this will fly, and whether we got the details right". It would be very good if this could be avoided by more careful deliberations and work up-front; the consequences of late changes for both security and implementers would be really bad.

If this is an extension, I'd personally prefer this to be in a separate document rather than to be part of TR #9.

Other Changes to the Bidi Algorithm
===================================

With the exception of minor tweaks, the bidi algorithm stayed stable since almost 15 years. But in recent years, there has been increased activity with new ideas for modification, both in the bidi algorithm itself and on higher levels (see the HTML work initiated by Aharon Lanin). It looks like these changes are being added piecemeal without yet seeing a new horizon of stability (after their IUC talk on Tuesday morning, people from Microsoft said that their parenthesis detection solution solved 13% of reported bidi problems; that means there may easily be more fixes comming).

But the bidi algorithm isn't an area where constant tinkering is advisable. It would therefore be very important that all these new initiatives are carefully checked against each other, and coordinated both in timing and in substance. It may be well advisable to wait with some of them so that many changes can be made 'in bulk' (the idea of an UBA 2.0), which will also help implementers.

Readability and Self-Containedness of the Document
==

In order to gain valuable comments not only from total insiders, the document has to be much more accessible to potential commenters. This starts with the title and the start of the introduction, which explicitly should mention email addresses and filenames, because it is otherwise ignored by people interested in these items.

The number of examples is extremely low (3). There are no examples of email addresses or filenames. There are no examples of non-generic (opaque syntax) URI schemes (e.g. mailto:,...). There are way too few examples to show what happens under different combinations of RTL and LTR components. There are no examples with realistic names (e.g. existing RTL top-level domains). There is a need for these to give people an everyday feel for the issue, while there is also a need to use abstract names (abc,...) to test usability when guessing is hard.

[The IRI spec, RFC 3987, has 10 examples (see http://tools.ietf.org/html/rfc3987#section-4.4) just to explain a single solution to the problem.]

All examples use the "uppercase is RTL" convention, which is good for outsiders, but doesn't show the potential end result for the people really affected. Parallel examples in Arabic and Hebrew are very important.

[As an RFC (all US-ASCII), the IRI spec was not able to include Arabic or Hebrew, but we made sure we provided Arabic and Hebrew equivalents for the examples (see http://www.w3.org/International/iri-edi ... mples.html) and referenced them from the spec. The 11th example has been added based on feedback. These examples are generated by a Ruby script, it should not be too difficult to change the script to produce examples for this spec.]

Security
========

The document correctly notes that ambiguous displays of bidi IRIs,... can cause security problems. However, the document is wrong and/or misleading in stating and/or implying that the proposal will remove ambiguity and confusion, except potentially in the very long term (10 to 20 years). The current specification for the display of bidi IRIs (RFC 3987, Section 4) uses the current bidi algorithm applied in an LTR context. In current implementations, display in an RTL context may also happen. A new specification will introduce at least a third alternative. While it may help reduce tinkering by implementers, it still creates (at least) one more alternative, and this should be very, very clearly noted in the document.

The document doesn't contain a security section, but it very clearly needs one. The IETF has an RFC on how to write good security sections.

Terminology
===========

The document uses 'fields' for e.g. individual domain name labels and path components. In the IETF, we have used 'component' for this; please align.

'surrogates' are mentioned as terminating characters. Are these surrogate pairs (in which case, it would be better to talk about non-BMP characters, but then it's totally unclear why these would terminate IRIs). Or are these unpaired surrogate units? In that case, I do not think the document should in any way prescribe how to handle stuff that is below the level of characters as codepoints. Otherwise, we would have to talk about incomplete UTF-8 byte sequences,...

BNF, Syntax Issues
==================

The document uses an ad-hoc and/or undefined syntactical notation. It says "This BNF uses a Perl-style syntax". Googling for "Perl-style" and "BNF" only leads to irrelevant stuff and the document itself. Please provide the syntax in a well-defined (with reference and syntax-checker, like e.g. the IETF ABNF) meta-syntax.

The meta-syntax uses so-called "smart" quotes. This has to be fixed.

Some non-terminals in the syntax are not defined. An example is <scheme>. Another is <percentEncodedUTF8>.

Some non-terminals use names different from those in the IRI spec although they are exactly the same. An example may be <percentEncodedUTF8>. This seems to correspond to <pct-encoded> in the IRI spec. If it doesn't, then the difference may be that it assumes an underlying UTF-8 encoding; such an assumption would be wrong, <ptc-encoded> can be used to represent raw bytes both in URIs and in IRIs.

The document only deals with the so-called "generic" syntax of IRIs. It always requires a double slash and a domain name after the scheme. However, many schemes do not use the "generic" syntax. An example is the mailto scheme; mailto:user@domain.tld would not be matched by the algorithm.

The document doesn't allow <iuserinfo> and <iport> components in the <iauthority> part (where it simply uses <domain>). Why were they excluded? Including additional syntax won't lead to many more false positives (because such strings look even more like IRIs than those without these components) and will avoid some false negatives.

With respect to potentially syntactically significant characters (i.e. all ASCII symbols), the document uses an approach completely different from the IRI spec, which makes checking of differences nearly impossible. Substractions in character classes are particularly confusing.

The use of character classes, in particular [[:L:][:N:][:M:][:S:][:Pd:][:Pc:][:Cf:]..., makes the syntax unreadable except to a very small set of regexperts, which have only a small overlap with Bidi and Uri experts. The IRI spec above ASCII excludes extremely little (just C1, the surrogate area, and non-characters, even private characters are allowed in query parts). It is unclear from the above cryptic syntax what is excluded, and why, and in asmuch as rare stuff is excluded, this doesn't really help making the extraction more precise.

There should be a complete list of ASCII symbol characters with their role/function in the IRI spec and in this spec. This is the best way to check for completeness. As an example, in the current syntax, "-" and "~" don't appear anywhere. Are they supposed to be included or excluded?

The IDN Label separators from IDNA 2003 are included despite the fact that they are not relevant in IDNA 2008 and they have never been allowed in IRIs. These definitely do appear in practice, but how often will they appear in IRIs involving RTL? My guess is that this chance is extremely low. If I had to cut corners, this is one instance where I'd do so; if somebody really cares about correct bidi display of an IRI with both RTL and ideographs, they should be able to use simple dots.

Related, the use of UTS46 probably offers too much leeway. Some restriction, e.g. in the symbol area (and in the area of compatibility characters), could bring some benefits for detection. After all, the overlap between leftovers from IDNA 2003 vanity symbol domain names and bidi-containing domain names can be assumed to be vanishingly small.

The <domain> rules allow a label separator at the end. This is technically correct, and allowed in URIs and IRIs (which don't deal at all with the internal structure of domain names, because in their place, names from other registry mechanisms could also be used). However, my guess is that a label separator at the end in vanishingly small in practice these days, and it might help excluding them for better precision.

The termination criterion includes unassigned (see also below re. dynamic updates), surrogates (see also above re. terminology), private-use, and control-code (what is meant by that exactly? C0+C1, or something else?) characters. My guess is that except the control codes, this really doesn't help much. Unassigned characters are by definition not used.

The explanation of the extraction/termination of IRIs is a mess. This is a place where an algorithmic description will help most. E.g. something along the lines of:
For detecting all IRIs in a given text, repeatedly scan for the first place where the IRI syntax matches, and take the longest match. Remove any final characters from that longest match to obtain a matched IRI, and continue detecting from the character immediately following the longest match.
(I'm not sure I got the details right (e.g. does only one dot get removed at the end, or two if there are two,...?), but that's the style I'd like to see here, because then I'd actually understand what's supposed to go on.)

RTL (and other non-ASCII) scheme names/alternates are clearly not allowed at this time, and there are no plans at all to introduced them. However, it would be prudent in my opinion to
a) explore how the various solutions work if ever RTL schemes are considered, and
b) if possible to define the algorithm so that it continues to work even in the event that they are introduced, rather than having to go through an additional revision.

The filename syntax doesn't include the very common Windows drive letter syntax.

There should be a list of syntactic differences between this spec and the IRI spec, with explanations, so that readers can jugde each difference on its merit rather than have to spend their time chasing details.

The spec seems to give some special status to some Latin-1 symbols (inverted exclamation mark, middle dot, inverted question mark). It is totally unclear why. The IRI spec is very clear that only ASCII symbols can take syntactic roles (there is no difference here between URIs and IRIs), and if there is some reason to include other symbol-like characters at some point in the syntax, there are clearly many many more such characters than just those in Latin-1.

Dynamic Updates?
================

The use of the list of top level domains at IANA is interesting because it provides quite some help to separate IRIs from non-IRIs. However, it is unclear whether the general expectation is that software should be dynamically updated with the IANA list, or whether it's okay to have longer release cycles. ICANN is apparently increasing the number of registrations per year, and many non-ASCII TLDs still remain to be defined. This means that with longer release cycles (e.g. smaller pieces of software that don't have a built-in update mechanism) in the mix, there will always be some discrepancy. This will create a highly undesirable long delay from registration to wide usability of a new TLD.

A similar issue appears with unassigned characters that are used as a termination criterion. These also will change from Unicode version to version.

Orders
======

http://tools.ietf.org/agenda/79/slides/iri-0.pdf, presented (remotely) at IETF 79, contains slides 19-23. In particular, slide 23 shows four possible solutions. Solution #2 on that slide is equivalent to Option 1 in the document under review. Options 2, 3, and 4 are essentially context/content-dependent variable choices from the table on slide 23.
(Similar kinds of overview tables may make this document way more easy to understand.)

The paragraph mentioning "big-endian" order in Option 1 is quite irrelevant. Users who are used to some given sequence of components and want to either see that sequence preserved (keep component order strictly LTR) or converted to their preferred directionality (change to have component order strictly go RTL) don't necessarily care about ultimate logic at all.

Option 1 has the disadvantage that even IRIs with RTL components only can use an LTR component order, which seems quite unnatural.

At the Unicode conference, on Tuesday morning, the group from Microsoft explained that preference for component order was not uniform, and not context- or content-dependent, but depended on country: Israel strongly preferred LTR component order, while many (but not all) Arabic countries preferred RTL order. According to their words, the situation was similar to what happens in Math, but there was no 100% correlation.

Regards, Martin.

! ? ⃠ ×

Top profile pm email edit quote

asmus

offline

Unicode Guru

Joined: Tue Dec 01, 2009 11:49 am
Posts: 114

Post subject: Re: October 2011 feedback on PRI 185 (long!)
Posted: Mon Oct 24, 2011 2:02 pm

> **MartinJD wrote:**
>
> These are my comments on "Extension of UBA for improved display of URL/IRIs", available from http://www.unicode.org/review/pri185/, as modified on Sept. 22 (presumably 2011).

I have inserted some of my comments, the quotes are selective.

> **MartinJD wrote:**
>
> Preexisting Specs and Parallel Work
> ==================================
>
> The IRI specification (RFC 3987, http://tools.ietf.org/html/rfc3987) as well as draft updates (http://tools.ietf.org/html/draft-ietf-iri-3987bis), and the specifications for Email Address Internationalization as well as their draft updates should be referenced at the start of the document. (There are no specs for file names as far as I know.) It's a good idea to point readers to more introductory material such as Richard Ishida's "idn-and-iri", but that's not enough for what is supposed to become (part of) a spec itself.
>
> RFC 3987 contains a section about the display of bidirectional IRIs (Section 4, http://tools.ietf.org/html/rfc3987#section-4.4). This should clearly be mentioned in the document. The section was written based on the following goals/assumptions:
>
> 1) That it would be desirable that bidi IRIs were displayed the same everywhere, both in places where they are identified as such (e.g. a browser's address/location bar) and in free text where no special processing could be applied to them.

> 2) That it was unfeasible to change the Unicode Bidirectional Algorithm (UBA) to deal with IRIs as a special case.
>
> The first assumption is shared by the current proposal; removing the second assumption is at the base of the current proposal.
>
> Now that changing (or extending) the UBA is on the table, we have to check what needs specifying, and where. My current take is that we have the following pieces:
>
> 1) Display of bidi IRIs once identified: UBA extension, with strong input from stakeholders in affected regions and from IRI WG.
>
> 2) Identifying IRIs in contexts: This would ideally be provided by the IETF. There is Appendix C of the URI spec (http://tools.ietf.org/html/rfc3986#appendix-C), Delimiting a URI in Context, and there was at least one attempt to do something in this direction (see http://tools.ietf.org/html/draft-yoneya ... gnition-00), but no wider interest and no pressure for standardization (the functionality seemed to work well where needed (e.g. email programs) and minor differences in implementation seemed to hurt nobody). So there's a rather large chance that this remains for the UTC to do, although with strong input from the IETF.
>
> 3) Restrictions on strong directionality mixing for components such as domain name labels: This is done for IDNA in RFC 5893 (http://tools.ietf.org/html/rfc5893) and is being updated and adapted based on the RFC 5893 effort for IRIs in the IRI WG. Input from the bidi experts in the UTC is greatly appreciated.
>
> We should make sure that we have got something like the above "pieces of the puzzle" right before we get too much into specific technical details.

Agreed - also to be put on the table would be additional issues not necessarily specific to IRIs. This is a good time to deal with the accumulated experience around the bidi algorithm. Instead of a "quick fix" it's time to have the same level of deep deliberation as surrounded the creation of the original bidi algorithm.

> **MartinJD wrote:**
>
> Document Target
> ===============

> There is talk about this being an experimental extension. Care should be given to be extremely clear what these two words mean, in particular because I don't know any other cases where this has been done in an Unicode context.
>
> Extension seems to mean "the bidi algorithm can be used with or without this". This is desirable from an implementer's perspective, but not from a security perspective.
>
> Experimental seems to mean "we aren't really sure yet whether this will fly, and whether we got the details right". It would be very good if this could be avoided by more careful deliberations and work up-front; the consequences of late changes for both security and implementers would be really bad.
>
> If this is an extension, I'd personally prefer this to be in a separate document rather than to be part of TR #9.

- I support the idea that the "original" bidi algorithm needs to remain on the books and identifiable as such.
- I concur with the desire to have a comprehensive set of extensions, leading to a single new version which, after a bug-fixing phase, will be stable.
- In addition to IRI/URI and filenames, an extended bidi Algorithm needs to fix the embedding model and general handling of separator characters.
- I am agnostic on the publication mechanisms for this. As long as the extensions are clearly marked, they could be in the same or a different "physical" document. There are advantages to each.
- I object to using the version of the associated Unicode Standard to mark the break between "old" and "extended" UBA. There are an unknown number of "old" implementations, many of which are from time to time updated to new character repertoire. There's no way the UTC can control or police the existence of "old" UBA implementations updated to "newly added" characters (e.g. extended Arabic repertoire).
- My preferred approach for maximizing interoperability would be to tie the use of the "new" bidi algorithm to some protocol (for example have HTML require its use, in addition to whatever security prootocls). In that way, interoperating implementations would be able to better predict in which contexts it is safe to expect the other side to support the new extensions

I have no problem with creating an extended "beta" or "provisional" version of the extended UBA, if it is understood as being aimed at letting people experiment with various implementations. A rather well-defined timetable for finalizing and incorporating such "beta" input would be advisable. Altogether, this work should not be shoehorned into the versioning cycle or beta period of any particular Unicode version.

> **MartinJD wrote:**
>
> Other Changes to the Bidi Algorithm
> ==================================
>
> With the exception of minor tweaks, the bidi algorithm stayed stable since almost 15 years. But in recent years, there has been increased activity with new ideas for modification, both in the bidi algorithm itself and on higher levels (see the HTML work initiated by Aharon Lanin). It looks like these changes are being added piecemeal without yet seeing a new horizon of stability (after their IUC talk on Tuesday morning, people from Microsoft said that their parenthesis detection solution solved 13% of reported bidi problems; that means there may easily be more fixes coming).
>
> But the bidi algorithm isn't an area where constant tinkering is advisable. It would therefore be very important that all these new initiatives are carefully checked against each other, and coordinated both in timing and in substance. It may be well advisable to wait with some of them so that many changes can be made 'in bulk' (the idea of an UBA 2.0), which will also help implementers.

Couldn't agree more with these two points, as outlined above in my list of issues.
Having a well-defined process for UBA2.0 with it's own development cycle and beta review would perhaps help getting more input from implementers, such as the case with the MS input you cite.

> **MartinJD wrote:**
>
> Readability and Self-Containedness of the Document
> ==
>
> In order to gain valuable comments not only from total insiders, the document has to be much more accessible to potential commenter. This starts with the title and the start of the introduction, which explicitly should mention email addresses and filenames, because it is otherwise ignored by people interested in these items.
>
> The number of examples is extremely low (3). There are no examples of

> email addresses or filenames. There are no examples of non-generic (opaque syntax) URI schemes (e.g. mailto:,...). There are way too few examples to show what happens under different combinations of RTL and LTR components. There are no examples with realistic names (e.g. existing RTL top-level domains). There is a need for these to give people an everyday feel for the issue, while there is also a need to use abstract names (abc,...) to test usability when guessing is hard.
>
> [The IRI spec, RFC 3987, has 10 examples (see http://tools.ietf.org/html/rfc3987#section-4.4) just to explain a single solution to the problem.]
>
> All examples use the "uppercase is RTL" convention, which is good for outsiders, but doesn't show the potential end result for the people really affected. Parallel examples in Arabic and Hebrew are very important.
>
> [As an RFC (all US-ASCII), the IRI spec was not able to include Arabic or Hebrew, but we made sure we provided Arabic and Hebrew equivalents for the examples (see http://www.w3.org/International/iri-edi ... mples.html) and referenced them from the spec. The 11th example has been added based on feedback. These examples are generated by a Ruby script, it should not be too difficult to change the script to produce examples for this spec.]

Those points are well-taken. In light of the desire to turn this into a more comprehensive effort, a structured document would be needed that can hold together the related areas proposed for change.
I think it's time to draft a comprehensive UBA 2.0 document in which to collect all the proposed or contemplated extensions, such as handling embeddings and separator characters in a more consistent way.

Of the existing bidi extension, solely the ALM (Arabic letter mark) can be moved forward as is, because it can be hadled by a simple, compatible repertoire extension (fully supportable by any "old" implementation that is updated with more recent property table).

____
A./

! ? ⊘ ×

Top profile pm email edit quote

Display posts from previous: All posts Sort by Post time Ascending Go

newtopic postreply **Page 1 of 1** [2 posts]

Board index » Public Review Discussions » PRI 185 - Extension of UBA for improved display of URL/IRIs

All times are UTC - 8 hours

Who is online

Users browsing this forum: **Sarasvati** and 0 guests

Quick-mod tools: Lock topic Go

You **can** post new topics in this forum
You **can** reply to topics in this forum
You **can** edit your posts in this forum
You **can** delete your posts in this forum
You **can** post attachments in this forum

Search for:
Go

Jump to: PRI 185 - Extension of UBA for improved display of URL/IRIs Go

[Administration Control Panel]