Internationalized Resource Identifiers   M. Duerst
(iri)   Aoyama Gakuin University
Internet-Draft   L. Masinter
Intended status: BCP   Adobe
Expires: April 24, 2013   A. Allawi
                                                  Diwan Software Limited
                                                     October 21, 2012

       **Guidelines for Internationalized Resource Identifiers with Bi-
                    directional Characters (Bidi IRIs)
                    draft-ietf-iri-bidi-guidelines-03**

Abstract

   This specification gives guidelines for selection, use, and
   presentation of International Resource Identifiers (IRIs) which
   include characters with inherent right-to-left (rtl) writing
   direction.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Overview

   Some UCS characters, such as those used in the Arabic and Hebrew
   scripts, have an inherent right-to-left (rtl) writing direction as
   opposed to characters, such as those in the Latin script, that have
   an inherent left-to-right (ltr) direction. IRIs containing rtl
   characters (called bidirectional IRIs or Bidi IRIs) require
   additional attention because of the non-trivial relation between
   their logical and visual ordering. The logical order represents the
   order in which characters are stored on computers and read by people.
   The visual order is the order in which the characters appear (or are
   expected to appear) on a computer display or printout.

   Generally, alphabetic characters in scripts like Arabic and Hebrew
   are drawn rtl while numbers are drawn ltr. Symbols such as slash
   ('/') and period ('.') take their visual direction from the
   surrounding characters. A list of all ASCII symbols with their
   bidirectional character type and their function in URIs and IRIs is
   given in Appendix A.

   Because of this complex interaction between the logical
   representation, the visual representation, and the syntax of a Bidi
   IRI, a balance is needed between various requirements. The main
   requirements are:

   1. user-predictable conversion between visual and logical
      representation;

   2. the ability to include a wide range of characters in various parts
      of the IRI; and

   3. minor or no changes or restrictions for implementations.

1.2. Availability

   This document is available in (line-printer ready) plaintext ASCII
   and in PDF. It is also available in HTML from
   http://www.sw.it.aoyama.ac.jp/2012/pub/
   draft-ietf-iri-bidi-guidelines-03.html, and in UTF-8 plaintext from
   http://www.sw.it.aoyama.ac.jp/2012/pub/
   draft-ietf-iri-bidi-guidelines-03.utf8.txt. While all these versions
   are identical in their technical content, the HTML, PDF, and UTF-8
   plaintext versions show non-Unicode characters directly. This often
   makes it easier to understand examples, and readers are therefore
   strongly advised to consult one of these versions in preference to or
   as a supplement to the ASCII version.

[1.3](). Notation

In this document, "Bidi Notation", abbreviated "BN" is used for the given Bidi IRI examples as follows: Lower case letters a-z stand for characters that are written with a left to right ordering (such as Latin characters), whereas upper case letters A-Z represent characters that are written right to left (such as Arabic or Hebrew characters). Numbers and symbols are the same.

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [[RFC2119]().

[2](). Logical Storage and Visual Presentation

When stored or transmitted in digital representation, Bidi IRIs MUST be in full logical order and MUST conform to the IRI syntax rules (which includes the rules relevant to their scheme). This ensures that Bidi IRIs can be processed in the same way as other IRIs.

Bidi IRIs MUST be visually ordered by the Unicode Bidirectional Algorithm [[UNIV6](], [[UNI9]). Bidi IRIs MUST be rendered in the same way as they would be if they were in a left-to-right embedding.

In conformance with the Unicode Bidirectional Algorithm, embedding MAY be done in one of two ways:

1. precede the IRI with U+202A, LEFT-TO-RIGHT EMBEDDING (LRE), and follow with U+202C, POP DIRECTIONAL FORMATTING (PDF); or

2. use a higher-level protocol (e.g., the dir='ltr' attribute in HTML).

Preceding and following the Bidi IRI with U+200E, LEFT-TO-RIGHT MARK (LRM) is NOT RECOMMENDED as, there are cases where this may not be sufficient to match full left to right embedding.

There is no requirement to use embedding if the display is still the same without the embedding. For example, a Bidi IRI in a text with left-to-right base directionality (such as used for English or Cyrillic) that is preceded and followed by whitespace and strong left-to-right characters does not need an embedding. Also, a bidirectional relative IRI reference that only contains strong right-to-left characters and weak characters (such as symbols) and that starts and ends with a strong right-to-left character and appears in a text with right-to-left base directionality (such as used for Arabic or Hebrew) and is preceded and followed by whitespace and

strong characters does not need an embedding.

However, implementers are RECOMMENDED to use embedding in all cases
where they are not completely sure that the display behavior is
unaffected without the embedding.

The Unicode Bidirectional Algorithm ([UNI9], section 4.3) permits
higher-level protocols to influence bidirectional rendering. Such
changes by higher-level protocols MUST NOT be used if they change the
rendering of IRIs.

The bidirectional formatting characters that may be used before or
after the IRI to ensure correct display are not themselves part of
the IRI. IRIs MUST NOT contain bidirectional formatting characters
(LRM, RLM, LRE, RLE, LRO, RLO, and PDF). They affect the visual
rendering of the IRI but do not appear themselves. It would
therefore not be possible to input an IRI with such characters
correctly.

3. Bidi IRI Structure

The Unicode Bidirectional Algorithm is designed for general purpose
text. To make sure that it does not affect the rendering of Bidi
IRIs outside of the requirements of this document, some restrictions
on Bidi IRIs are necessary. These restrictions are given in terms of
delimiters (structural characters, mostly punctuation such as "@",
".", ":", and "/") and components (usually consisting mostly of
letters and digits).

The following syntax rules from the ABNF of [RFC3987bis] correspond
to components for the purpose of Bidi behavior: iuserinfo, ireg-name,
isegment, isegment-nz, isegment-nz-nc, ireg-name, iquery, and
ifragment.

Specifications that define the syntax of any of the above components
MAY divide them further and define smaller parts to be components
according to this document. As an example, the restrictions of
[RFC3490] on bidirectional domain names correspond to treating each
label of a domain name as a component for schemes with ireg-name as a
domain name. Even where the components are not defined formally, it
may be helpful to think about some syntax in terms of components and
to apply the relevant restrictions. For example, for the usual name/
value syntax in query parts, it is convenient to treat each name and
each value as a component. As another example, the extensions in a
resource name can be treated as separate components.

For each component, the following restrictions apply:

1. A component SHOULD NOT use both right-to-left and left-to-right
   characters.

2. A component using right-to-left characters SHOULD start with a
   right-to-left character, and end with a right-to-left character
   potentially followed by one or more nonspacing mark (bidi class
   NSM).

The above restrictions are given as "SHOULD"s, rather than as
"MUST"s. For IRIs that are never presented visually, they are not
relevant. However, for IRIs in general, they are very important to
ensure consistent conversion between visual presentation and logical
representation, in both directions.

Note: In some components, the above restrictions may actually be
   strictly enforced. For example, [RFC3490] requires that these
   restrictions apply to the labels of a host name for those schemes
   where ireg-name is a host name. In some other components (for
   example, path components) following these restrictions may not be
   too difficult. For other components, such as parts of the query
   part, it may be very difficult to enforce the restrictions because
   the values of query parameters may be arbitrary character
   sequences.

If the above restrictions cannot be satisfied otherwise, the affected
component can always be mapped to URI notation using the general
percent-encoding of IRI components, as described in [RFC3987bis].
Please note that the whole component has to be mapped (see also
Example 9 below).

4. Input of Bidi IRIs

Bidi input methods MUST generate Bidi IRIs in logical order while
rendering them according to Section 2. During input, rendering
SHOULD be updated after every new character is input to avoid end-
user confusion.

5. Examples

This section gives examples of Bidi IRIs in Bidi Notation. It shows
legal IRIs with the relationship between their logical and visual
representation and explains how certain phenomena in this
relationship may look strange to somebody not familiar with
bidirectional behavior, but familiar to users of Arabic and Hebrew.
It also shows what happens if the restrictions given in Section 3 are
not followed. Please see <Availability> for versions of the examples

in Arabic and Hebrew script.

To read the bidi text in the examples, read the visual representation
from left to right until you encounter a block of rtl text. Read the
rtl block (including slashes and other special characters) from right
to left, then continue at the next unread ltr character.

Please note that "BN" stands for "Bidi Notation", see <Notation>. AR
stands for Arabic, HE for Hebrew.

Example 1: A single component with rtl characters is inverted:
Logical representation (BN): "http://ab.CDEFGH.ij/kl/mn/op.html"
Visual representation (BN): "http://ab.HGFEDC.ij/kl/mn/op.html"
Components can be read one by one, and each component can be read in
its natural direction.

Example 2: More than one consecutive component with rtl characters is
inverted as a whole:
Logical representation (BN): "http://ab.CDE.FGH/ij/kl/mn/op.html"
Visual representation (BN): "http://ab.HGF.EDC/ij/kl/mn/op.html"
A sequence of rtl components is read rtl, in the same way as a
sequence of rtl words is read rtl in a bidi text.

Example 3: All components of an IRI (except for the scheme) are rtl.
All rtl components are inverted overall:
Logical representation (BN):
"http://AB.CD.EF/GH/IJ/KL?MN=OP;QR=ST#UV"
Visual representation (BN): "http://VU#TS=RQ;PO=NM?LK/JI/HG/FE.DC.BA"
The whole IRI (except the scheme) is read rtl. Delimiters between
rtl components stay between the respective components; delimiters
between ltr and rtl components don't move.

Example 4: Each of several sequences of rtl components is inverted on
its own:
Logical representation (BN): "http://AB.CD.ef/gh/IJ/KL.html"
Visual representation (BN): "http://DC.BA.ef/gh/LK/JI.html"
Each sequence of rtl components is read rtl, in the same way as each
sequence of rtl words in an ltr text is read rtl.

Example 5: Example 2, applied to components of different kinds:
Logical representation (BN): "http://ab.cd.EF/GH/ij/kl.html"
Visual representation (BN): "http://ab.cd.HG/FE/ij/kl.html"
The inversion of the domain name label and the path component may be
unexpected, but it is consistent with other bidi behavior. For
reassurance that the domain component really is "ab.cd.EF", it may be
helpful to read aloud the visual representation following the Unicode
Bidirectional Algorithm. After "http://ab.cd." one reads the RTL
block "E-F-slash-G-H", which corresponds to the logical

representation.

Example 6: Same as Example 5, with more rtl components:
Logical representation (BN): "http://ab.CD.EF/GH/IJ/kl.html"
Visual representation (BN): "http://ab.JI/HG/FE.DC/kl.html"
The inversion of the domain name labels and the path components may
be easier to identify because the delimiters also move.

Example 7: A single rtl component includes digits:
Logical representation (BN): "http://ab.CDE123FGH.ij/kl/mn/op.html"
Visual representation (BN): "http://ab.HGF123EDC.ij/kl/mn/op.html"
Numbers are written ltr in all cases but are treated as an additional
embedding inside a run of rtl characters. This is completely
consistent with usual bidirectional text.

Example 8 (not allowed): Numbers are at the start or end of an rtl
component:
Logical representation (BN): "http://ab.cd.ef/GH1/2IJ/KL.html"
Visual representation (BN): "http://ab.cd.ef/LK/JI1/2HG.html"
The sequence "1/2" is interpreted by the Bidirectional Algorithm as a
fraction, fragmenting the components and leading to confusion. There
are other characters that are interpreted in a special way close to
numbers; in particular, "+", "-", "#", "$", "%", ",", ".", and ":".

Example 9 (not allowed): The numbers in the previous example are
percent-encoded:
Logical representation (BN): "http://ab.cd.ef/GH%31/%32IJ/KL.html"
Visual representation (BN): "http://ab.cd.ef/LK/JI%32/%31HG.html"

Example 10 (allowed but not recommended):
Logical representation (BN): "http://ab.CDEFGH.123/kl/mn/op.html"
Visual representation (BN): "http://ab.123.HGFEDC/kl/mn/op.html"
Components consisting of only numbers are allowed (it would be rather
difficult to prohibit them), but these may interact with adjacent RTL
components in ways that are not easy to predict.

Example 11 (allowed but not recommended):
Logical representation (BN): "http://ab.CDEFGH.123ij/kl/mn/op.html"
Visual representation (BN): "http://ab.123.HGFEDCij/kl/mn/op.html"
Components consisting of numbers and left-to-right characters are
allowed, but these may interact with adjacent RTL components in ways
that are not easy to predict.

6. IANA Considerations

This document makes no changes to IANA registries.

7. Security Considerations

   Confusion can occur with bidirectional IRIs, if the restrictions in
   Section 3 are not followed. The same visual representation may be
   interpreted as different logical representations, and vice versa. It
   is also very important that a correct Unicode bidirectional
   implementation be used.

8. Acknowledgements

   This document was derived from [RFC3987] and [RFC3987bis] and the
   acknowledgments of those documents apply. Shunsuke Oshima provided
   the data for Appendix A.

9. Main Changes Since RFC 3987

   This section describes the main changes since [RFC3987].

   o Separated out the section on bidi in [RFC3987] to this document.

   o Added examples in Arabic and Hebrew, which can be seen in html/
      pdf/utf8.txt versions.

   o Allowed NSMs at the end of components, for Dhivehi, Yiddish,...

   o TODO: check for major changes between RFC3987 and draft -02.

   Note to RFC Editor: Please remove this paragraph before publication.
   Detailed change logs are available in the IETF tools subversion
   repository at http://trac.tools.ietf.org/wg/iri/trac/log/
   draft-ietf-iri-3987bis/draft-ietf-iri-bidi-guidelines.xml.

10. References

10.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC3490] Faltstrom, P., Hoffman, P., and A. Costello,
              "Internationalizing Domain Names in Applications (IDNA)",
              RFC 3490, March 2003.

   [RFC3987bis]
              Duerst, M., Masinter, L., and M. Suignard,

                    "Internationalized Resource Identifiers (IRIs)",
                    October 2012,
                    <http://tools.ietf.org/id/draft-ietf-iri-3987bis>.

   [UNI9]   Davis, M., "The Unicode Bidirectional Algorithm", Unicode
                Standard Annex #9, September 2012,
                <http://www.unicode.org/reports/tr9/tr9-27.html>.

   [UNIV6]   The Unicode Consortium, "The Unicode Standard, Version
                6.2.0 (Mountain View, CA, The Unicode Consortium, 2012,
                ISBN 978-1-936213-07-8)", October 2012.

10.2. Informative References

   [RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource
              Identifiers (IRIs)", RFC 3987, January 2005.

Appendix A. List of ASCII Symbols and their Bidirectional Character Types

   To help understand the influence of various symbols on IRI display,
   this appendix lists all of them, giving the character itself, the
   Unicode codepoint, the character name, the bidirectional character
   type (BCT) and the rule and relevance in the IRI syntax.

   The most important ones in practice are ":", delimiting schem and
   port (CS, Common Number Separator), "/" to indicate generic
   (hierarchical) schemes and as a path separator (CS, Common Number
   Separator), "?" to introduce a query part (ON, Other Neutral), "#" to
   introduce a fragment identifier (ET, European Number Terminator), "."
   to separate labels in a domain name (CS, Common Number Separator),
   "&" to separate form parameters (ON, Other Neutral), and "@" to
   separate user information (ON, Other Neutral).

```
   Char Codepoint  Character Name       BCT  IRI syntax
   ----------------------------------------------------------------
   "#"  U+0023     NUMBER SIGN          ET   gen-delims, fragments
   "/"  U+002F     SOLIDUS              CS   gen-delims, paths
   ":"  U+003A     COLON                CS   gen-delims, scheme, port
   "?"  U+003F     QUESTION MARK        ON   gen-delims, query part
   "@"  U+0040     COMMERCIAL AT        ON   gen-delims, user
   "["  U+005B     LEFT SQUARE BRACKET  ON   gen-delims
   "]"  U+005D     RIGHT SQUARE BRACKET ON   gen-delims
   "%"  U+0025     PERCENT SIGN         ET   pcd-encoded
   "!"  U+0021     EXCLAMATION MARK     ON   sub-delims
   ","  U+002C     COMMA                CS   sub-delims
   "+"  U+002B     PLUS SIGN            ES   sub-delims
   "$"  U+0024     DOLLAR SIGN          ET   sub-delims
   "("  U+0028     LEFT PARENTHESIS     ON   sub-delims
   "'"  U+0027     APOSTROPHE           ON   sub-delims
   ")"  U+0029     RIGHT PARENTHESIS    ON   sub-delims
   "*"  U+002A     ASTERISK             ON   sub-delims
   ";"  U+003B     SEMICOLON            ON   sub-delims
   "="  U+003D     EQUALS SIGN          ON   sub-delims, forms
   "&"  U+0026     AMPERSAND            ON   sub-delims, forms
   "."  U+002E     FULL STOP            CS   unreserved, domain names
   "-"  U+002D     HYPHEN-MINUS         ES   unreserved
   "_"  U+005F     LOW LINE             ON   unreserved
   "~"  U+007E     TILDE                ON   unreserved
   " "  U+0020     SPACE                WS   excluded, delim
   '"'  U+0022     QUOTATION MARK       ON   excluded, delim
   "\"  U+005C     REVERSE SOLIDUS      ON   excluded, unwise
   "^"  U+005E     CIRCUMFLEX ACCENT    ON   excluded, unwise
   "<"  U+003C     LESS-THAN SIGN       ON   excluded, delim
   ">"  U+003E     GREATER-THAN SIGN    ON   excluded, delim
   "`"  U+0060     GRAVE ACCENT         ON   excluded, unwise
   "|"  U+007C     VERTICAL LINE        ON   excluded, unwise
   "{"  U+007B     LEFT CURLY BRACKET   ON   excluded, delim
   "}"  U+007D     RIGHT CURLY BRACKET  ON   excluded, delim
```

Authors' Addresses

   Martin J. Duerst (Note: Please write "Duerst" with u-umlaut wherever
                 possible, for example as "Dürst" in XML and HTML.)
   Aoyama Gakuin University
   5-10-1 Fuchinobe
   Chuo-ku
   Sagamihara, Kanagawa 252-5258
   Japan

   Phone: +81 42 759 6329
   Fax: +81 42 759 6495
   Email: duerst@it.aoyama.ac.jp
   URI: http://www.sw.it.aoyama.ac.jp/D%C3%BCrst/
          (Note: This is the percent-encoded form of an IRI)

   Larry Masinter
   Adobe
   345 Park Ave
   San Jose, CA 95110
   U.S.A.

   Phone: +1-408-536-3024
   Email: masinter@adobe.com
   URI: http://larry.masinter.net

   Adil Allawi
   Diwan Software Limited
   37-39 Peckham Road
   London SE5 8UH
   United Kingdom

   Phone: +44 7718 785850
   Fax: +44 20 72525444
   Email: adil@diwan.com
   URI: http://ironymark.diwan.com/