

draft-ietf-iri-bidi-guidelines-02

Issues Overview

IETF 83, Paris
IRI WG Meeting
2012-03-30

Martin J. Dürst, co-Editor

Overview

- Overview/Background/Goals
- Open Issues:
 - Editorial
 - Character-related
 - Serious
 - Procedural

Bidi(rectionality) Basics

- Arabic, Hebrew,... scripts read TFEL2THGIR
(in examples, we use ESAC REPPU for right-to-left)
- Storage is in logical order (parsing,... is easy)
- Display for general purpose text is specified by [Unicode TR 9](#)
 - Directionality of punctuation follows surrounding letters
 - In computer syntax, stuff gets thrown around

IRI Bidi Concepts

- Component: String between syntax characters
 - Domain name label
 - Path component
 - Query parameter name/value
 - ...
- Component directionality:
 - Each component clearly one way, to avoid letters jumping punctuation
- Run: Same-directionality component sequence

Bidi IRI Goals

1. User-expected logical \Leftrightarrow display conversion
2. Uniform logical \Leftrightarrow display conversion
3. Low implementation cost (ideally same as TR 9)
4. Allow wide range of character combinations

Problems:

- Goals conflict, can't have everything
- Right balance different from RFC 3987?

Bidi Open Issues: Editorial

- Issue #118: [What term to use for the kind of text that the Unicode Bidi Algorithm was designed for](#)
- [Proposal \(by Adil\)](#): "The Unicode Bidirectional Algorithm is designed for general purpose text"
- No disagreement on mailing list ⇒ implement
- Issue #116: [logical order and 'read' order](#)

Bidi Open Issues: Characters

Issue #28: [allow numbers at end of bidi components?](#)

- IDNA currently allows this, with restrictions
- We should not be more strict for IDNA
- Careful evaluation needed for component separators other than “.”

Bidi Open Issues: Characters

Issue #25: [Adapt rules for bidi components to those in IDNA2008](#)

IDNA2008 requirement:

- Label (\rightarrow component) uniqueness (1-to-1 logical \Leftrightarrow display conversion)
- Character grouping (don't jump the dot)
- Need to check for : / ? & = # , ...
- Harald's checking script (Perl) needs adaption
- Serious work

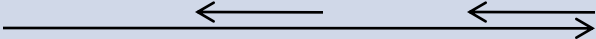

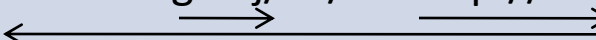
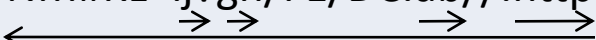
Bidi Open Issues: Serious

Issue #121: [Some users are requiring right-to-left label ordering](#)

- Many RTL users strongly prefer STCUDORP/MOC.abc.BEW//:http to http://BEW.abc.STCUDORP/MOC (RFC 3987) or http://BEW.abc.MOC/STCUDORP
- Easy for RTL-only
- Difficult to implement (IRI detection logic needed)
- Coordination with Unicode Consortium may be needed
- Preference depends on country, background, context
- Allow display preference variation in exchange for restriction on component sequence?
- Mixed cases may allow various spoofing attacks

Bidi IRI Ordering Alternatives

Logical: `http://ab.CD/EF/gh?ij=KL#MN`

Overall Directionality	Reordering by	Example	RFC 3987	Unicode TR #9	Users	#
LTR →	run	<code>http://ab.FE/DC/gh?ij=NM#LK</code> 	okay	possible	☹️	1
LTR →	component	<code>http://ab.DC/FE/gh?ij=LK#NM</code> 	bad	need exception	☹️	2
RTL ←	run	<code>NM#LK=gh?ij/FE/DC.http://ab</code> 	bad	possible	☹️	3
RTL ←	component	<code>NM#KL=ij?gh/FE/DC.ab//:http</code> 	bad	need exception	😊 ?	4

- Worst-case example, shows main design choices
- Conflict between users (and user-oriented vendors) and security concerns

Bidi Open Issues: Procedural

- Issue #117: [conformance requirements in bidi document -- do they belong?](#)
- Draft currently has 7 MUSTs and 3 SHOULDs
- Are these MUSTs/SHOULDs the right ones?
⇒ Issue #121
- Should we have MUSTs/SHOULDs?