Overview

• Overview/Background

• Issues in Groups:
  (discussion after each group)
  – Conversion by Decomposition
  – Registered Names
  – Query Parts
  – Legacy and Bugwards Compatibility
  – Bidirectionality
  – Weed-out
  – Other issues
Background

• IRI: Internationalized Resource Identifier, currently RFC 3987
• Internationalized (i.e. not-ASCII-only) version of URI (STD66, RFC 3986)
• Updating draft-ietf-iri-3987bis-03.txt
• List of open issues at: http://trac.tools.ietf.org/wg/iri/trac/report/1
• SVN revision log: http://trac.tools.ietf.org/wg/iri/trac/log/draft-ietf-iri-3987bis/draft-ietf-iri-3987bis.xml
IRI Examples

- http://zh.wikipedia.org/zh/中国互联网协会
- http://بوابة.تونس
Please Don’t Forget

• URIs/IRIs are a META-syntax
• Many pieces with different requirements get thrown together
• URIs/IRIs can be:
  – Absolute, complete from scheme to fragment id
  – Relative, just one or a few pieces
  – User-oriented (short, memorable)
  – Back-end (long, complicated)
Biggest Change from RFC 3987

• Conversion: IRI ⇒ URI
• Old: Conversion by Decomposition

IRI: Unicode Characters
(electrons, ink, air waves)

encode as UTF-8

Unicode in UTF-8

%-encode non-ASCII bytes

URI
Conversion: New

scheme://iregname/ipath1/ipath2/…/ipath.ext?iquery#ifragment

Decompose:
- scheme
- iregname
- ipath
- iquery
- ifragment

Convert:
- scheme: as-is
- iregname: punycode
- ipath: UTF-8 → %
- iquery: doc-enc → %
- ifragment: UTF-8 → %

Compose:
- scheme://regname/path1/path2/…/path.ext?query#fragment
New Conversion: Advantages

• Deal with special cases
  –  irez-name
  –  query Part
• Base for API (HTML5,...)
New Conversion: Status

• IRI ⇒ URI:
  – Basic writedown completed
  – Need to check details [issue #13; please help!]

• URI ⇒ IRI: TODO [issue #14]
Registered Names: Mapping to URIs

[issue #35]

• STD 66/RFC 3986: Allows %-encoding in reg-name
• RFC 3987: Allows conversion using %-encoding for ireg-name
• draft-03: MUST convert to punycode
Advantages of -03 Approach

• Less variability for URIs (only 中国 and xn--fiqs8s, not also %E4%B8%AD%E5%9B%BD)

• Better resolution: For IDNA, punycode always resolves

➢ Practical approach, but MUST too much (!?)
Disadvantages

• Conversion less uniform, more effort needed
• Does not deal with other systems (MDNS,..., see draft-iab-idn-encoding)
• Unnecessary restriction on schemes (IANA registry)
• Does not deal with opaque syntax, domain names in query part,...
• Interaction with mapping
• Forward-compatibility
• %-encoding is still legal
• Layer violation
Reg-name: Mapping among Unicode

• To map or not to map?
• When to map?
• How to map? (IDNA 2003? RFC 5895? TR 46?)

[issue #44]

• Proposal:
  – IRI creation: SHOULD only use U-Labels
  – Conversion to URI: MAY map (RFC 5895 or TR 46)
Query Part Encodings
[issues #24, #40]

• From an HTML <form> (GET request):
  – Document encoding [issue #11], or
  – Follow accept-charset attribute

• Input by user:
  – UTF-8: Preferred by big sites
  – Local user encoding: Preferred by some sites in Asia, non-interoperable

• In a document (e.g. <a src=...):
  – Document encoding
Query Part: Scheme Dependency

• Document encoding (where available):
  – http:/https:
  – What else? [Please help!]

• UTF-8:
  – mailto:
  – What else? [Please help!]

• Schemes without query part:
  – What? [Please help!]
Legacy and Bugwards Compatibility

• LEIRI: Legacy Extended IRI, for XML
  – Problem: Main XML specs diverged on an early draft of RFC 3987
  – Solution: Allow spaces,... in LEIRIs, with lots of health warnings
  – Status: One of the most carefully checked parts of the draft (Section 7.1) [issue #30]
Bugwards Compatibility: HTML5

• [issues #1, #2, #3]

• Browsers do a lot more than what the specs require

• Browser makers want to get the spec up to speed with reality
Bugwards Compatibility Examples

• Allow single '%'? [issue #41]
• Allow '#' in fragment part? [issue #42]
• Illegal IRI characters [issue #43]
• Many others, wide variance in implementations

• Section, appendix, separate draft?
• draft in preparation by Adam Barth
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 running text is specified by
  Unicode TR 9
  – Directionality of punctuation follows surrounding letters
  – In computer syntax, stuff gets thrown around
Bidi IRI Goals

• Easily readable (for native readers)
• Easy to display (ideally no deviation from TR 9)
• Consistent conversion logical ⇔ display
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 Issues

• Adapt Bidi character restrictions to IDNA2008 [issue #25]
  – Allow combining marks at end of component (no-brainer)
  – Allow digits at end of component (probably yes, issue #28)
  – Establish non-jumping restrictions for IRIs (needs work, please help)

• Overall display strategy

• Move to separate document? [issue #6]
### Bidi IRI Ordering Alternatives

<table>
<thead>
<tr>
<th>Overall Directionality</th>
<th>Reordering by</th>
<th>Example</th>
<th>RFC 3987</th>
<th>Unicode TR #9</th>
<th>Users</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTR →</td>
<td>run</td>
<td><a href="http://ab.FE/DC/gh?ij=NM#LK">http://ab.FE/DC/gh?ij=NM#LK</a></td>
<td>okay</td>
<td>possible</td>
<td>😞</td>
<td>1</td>
</tr>
<tr>
<td>LTR →</td>
<td>component</td>
<td><a href="http://ab.DC/FE/gh?ij=LK#NM">http://ab.DC/FE/gh?ij=LK#NM</a></td>
<td>bad</td>
<td>need exception</td>
<td>😞</td>
<td>2</td>
</tr>
<tr>
<td>RTL ←</td>
<td>run</td>
<td>NM#LK=gh?ij/FE/DC.<a href="http://ab">http://ab</a></td>
<td>bad</td>
<td>possible</td>
<td>😞</td>
<td>3</td>
</tr>
<tr>
<td>RTL ←</td>
<td>component</td>
<td>NM#KL=ij?gh/FE/DC.ab://http</td>
<td>bad</td>
<td>need exception</td>
<td>😞</td>
<td>4</td>
</tr>
</tbody>
</table>

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

• Section 6: Use of IRIs [Please help reviewing!]
• Section 8: URI/IRI Processing Guidelines (Informative) [Please help reviewing!]
• Security Section: Replace large parts by pointers to Unicode TR 36 [issue #18]
• Appendix A, Design Considerations: Replace with pointer to RFC 3987 [issue #53]
Other Issues (except trivial)

- **#5**: Distinguish IRI vs. "Presentation of IRI"?
- **#15**: Move comparision section to separate document?
- **#20**: Update Acknowledgements Section
- **#22**: Fix "IRIs as identity tokens MUST"
- **#23**: When to use NFC normalizing transcoder? (close?)
- **#26**: No combining marks at start of component?
- **#27**: Anything to say about ZWNJ/ZWJ?
- **#29**: Include tag ranges in <iprivate> (close?)
- **#34**: Incomplete sentence
- **#36**: Some HTTP implementations send UTF-8 paths
- **#39**: Warn about wrong conversion of non-BMP characters
- **#45**: Secure comparisons
- **#46, #47**: Length limits
- **#52**: Update reference to Unicode 6.0 (patch available)