Network Working Group	C. Weber
Internet-Draft	Casaba Security
Intended status: Standards Track	July 12, 2011
Expires: January 13, 2012	

Guidelines for Implementers of Internationalized Resource Identifiers (IRIs)
draft-weber-iri-guidelines-01

Abstract

Some members of the implementation community have expressed confusion about the rules and algorithms for processing Internationalized Resource Identifiers (IRIs). This document aims to clarify these matters and improve interoperability around IRI processing by summarizing the steps required to prepare and parse arbitrary Unicode strings as Internationalized Resource Identifiers. Further goals of this document are to define limited scheme-specific rules around IRI processing and to define the steps required for producing the canonical form of an IRI.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Internationalized Resource Identifiers (IRIs) extend the Uniform Resource Identifier (URI) specification [RFC3986] by opening up the authority, path, query, and fragment components to the character space available in Unicode/ISO 10646. Arbitrary Unicode strings may be prepared and parsed into IRI sub-components which map directly to the same URI sub-components.
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Terminology

This section defines terminology used in this document. Unicode characters are referred to using the notation described in section 3 of [RFC5137]. A Unicode code point is represented as U+NNNN where the NNNN string consists of the code point's hexadecimal numbers.
reference-string
The original and unprocessed input string being considered as an IRI reference.
pre-processed-reference-string
The reference-string that has been through the pre-processing steps.
relative reference
The term "relative reference" used in this document can be interpreted according to the rules in section 4.2 of [RFC3986] using the additionally allowed characters that [RFC3987] permits in each component.
percent-encode
Convert each octet of a sequence to %HH, where HH is the hexadecimal notation of the octet value.

3. Sources

This document makes reference to the following sources of information about the parsing of IRIs:

1: Internationalized Resource Identifiers (IRIs) as specified in http://tools.ietf.org/html/draft-ietf-iri-3987bis-05 [RFC3987]

2: Parsing URLs for Fun and Profit, http://tools.ietf.org/html/draft-abarth-url-01 [2]

3: HTML Living Standard: URLs, http://www.whatwg.org/specs/web-apps/current-work/multipage/urls.html#urls [3]

4:
Change proposal for ISSUE-56, http://lists.w3.org/Archives/Public/public-html/2010Jul/0036.html [4]

5: URIs, URLs, and URNs: Clarifications and Recommendations 1.0, http://www.w3.org/TR/uri-clarification/ [5]

6: HTML CHANGE PROPOSAL; change definition of URL to normative reference to IRIBIS, http://lists.w3.org/Archives/Public/public-html/2010Feb/0882.html [6]

7: BIDI URL Display, https://docs.google.com/document/d/1c8-svx7og0qBUfGBobw7LYfOcNeDVPYbNVMNpSqYCFo/edit?hl=en [7]

4. Pre-processing Arbitrary Unicode Strings

This section describes the pre-processing steps required to prepare an arbitrary Unicode reference-string for later parsing into IRI sub-components.

1. Remove leading and trailing instances of ASCII whitespace characters U+0020 SPACE, U+000D CARRIAGE RETURN (CR), U+000A LINE FEED (LF), and U+0009 CHARACTER TABULATION from the string. Note that Unicode has many more characters that are considered whitespace, none of which are affected or considered in this rule.
   Reference: Section 7.2 of [RFC3987] for details.

2. If more than one reference is allowed, split the string into substrings on blocks of contiguous U+0020 SPACE characters. Each of one of these substrings is an independent reference-string and will be processed individually. If more than one reference is not allowed, either remove blocks of contiguous whitespace or replace each U+0020 SPACE with a single percent-encoded U+0020 SPACE, written as "%20", depending on what is required for the current context.
   Reference: Mentioned in [4].

3. If the current string is not already in a Unicode encoding, then transcode the string to the a Unicode encoding such as UTF-8, UTF-16, or UTF-32.
   Reference: Section 3.1 of [RFC3987].

4. TODO: Should numerical character references be replaced with their corresponding character? e.g. <http://www.example.com/foobar/foo?bar> would become <http://www.example.com/foobar/foo?bar> during this step. Or should this step of unescaping be limited to the <iunreserved> set?

This is the pre-processed-reference-string ready for parsing.

5. Parsing Unicode Strings into IRI Components

With an arbitrary IRI string that has been through pre-processing, referred to as the "pre-processed-reference-string", this section describes the subsequent process of parsing the string into its five major IRI sub-components using rules defined by [RFC3896] (using an algorithm equivalent to Appendix B of [RFC3986]) but with updated ABNF of [RFC3987]. These rules are summarized here.
Reference: Section 3.2 of [RFC3987].

5.1. Identify the scheme

If the current string does not contain a ":" U+003A then the string does not contain a scheme and the pre-processed-reference-string may be handled as a relative reference according to the rules in section 4.2 of [RFC3986] using the additionally allowed characters that [RFC3987] permits.

*Continue to "Identify the path"

*Abort further scheme processing

If the first character of the string is not an <ALPHA> then this is not a valid scheme and the pre-processed-reference-string may be handled as a relative reference.

*Continue to "Identify the path"

*Abort further scheme processing

Consume all characters up to but not including the first occurrence of ":" U+003A. If the consumed substring contains any characters other than < ALPHA / DIGIT / "+" / "-" / "." > then it is not a valid scheme and the pre-processed-reference-string may be handled as a relative reference.

*Continue to "Identify the path"

*Abort further scheme processing

The consumed substring at this point is the scheme. Skip over the ":" U+003A character. Continue parsing the remaining string as an authority part.

5.2. Identify the authority

The URI authority component may contain userinfo, a host, and a port. If the current string does not begin with the two characters "//"

U+002F U+002F then the string is not an authority and may be handled as a path sub-component.

*Continue to "Identify the path"

*Abort further authority processing

Consume up to the first occurrence of any one of the authority's terminating characters "/" U+002F, "?" U+003F, "#" U+0023, or the end of the string. This is the authority, also known as the iauthority under IRI RFC3987. Continue further parsing of the authority to identify the userinfo, host, and port parts.

5.2.1. Identify the userinfo

The userinfo may come in the form of a username and password rendered as "user:password". The userinfo part may be parsed according to the rules of RFC3986 Section 3.2.1 with the updated ABNF for iuserinfo in RFC3987.
If the authority does not contain a "@" U+0040 then the string does not contain a userinfo part and the authority may be parsed for a host and port part.

*Continue to "Identify the host"

*Abort further userinfo processing

From the beginning of the authority, consume each character up to but not including the first occurrence of "@" U+0040. This is the userinfo. Further parsing of the userinfo is scheme-specific.
Skip over the first occurrence of "@" U+0040 following the userinfo, and continue parsing the remaining authority to identify the host.

5.2.2. Identify the host

The host part of authority may contain an IP-literal, IPv4address, or a reg-name according to the ABNF rules of RFC3986 updated to support Unicode characters in the ireg-name as described in RFC3987. Consume all characters up to but not including the last ":" U+003A character or the end of authority.
If this substring is determined to be an IP-literal or IPv4address, then the consumed characters are the host.

*If the authority did contain a ":" then continue parsing the remaining authority including the ":" character according to the "Identify the port" section

*Abort further host processing

Else the substring is determined to be an ireg-name according to the ABNF naming convention from RFC3987. This is the host.

*If the authority did contain a ":" then the remaining authority including the ":" character will be processed according to the "Identify the port".

*Continue processing the host

The host SHOULD be processed according to the rules of IDNA2008, but MAY be processed according to UTS46 or IDNA2003. If the host is in DNS Internet dot-notaion then it's labels SHOULD be converted to punycode.
This is the host.
TODO: Error handling. Mention leaving the host name in pure Unicode form for intranet/local name scenarios that don't use DNS, e.g. WINS?

5.2.3. Identify the port

Further processing SHOULD skip the first occurrence of ":" U+003A and consume the remaining characters. If these characters are not *DIGIT then the port is invalid.

*Continue to "Identify the path"

*Abort further scheme processing

Else this is the port.

5.3. Identify the path

Consume the remaining pre-processed-reference-string up to but not including the first occurrence of a terminating character "?" U+003F, "#" U+0023, or the end of the string.
Percent-encode all characters present from the ucschar list.
If the path contains any characters not allowed by the ABNF of RFC3987 Section 2.2 or Section 7.2 then replace those characters with their percent-encoding.
This is the path.
If the terminating character was "?" then process the remaining string including the leading "?" according to "Identify the query".
If the terminating character was "#" then process the remaining string including the leading "#" according to "Identify the fragment".
TODO: Handling of special characters "/" and "\".

5.4. Identify the query

Consume the remaining string starting with the leading "?" and up to but not including the first occurrence of "#" or the end of the string.
Percent-encode all characters present from the ucschar list.

If the path contains any characters not allowed by the ABNF of [RFC3987] Section 2.2 or the lists in Section 7.2 then replace those characters with their percent-encoding.
This is the query component.
If the terminating character was "#" then process the remaining string including the leading "#" according to "Identify the fragment".
TODO: Handling of special characters "&", "?", "=", and "/"

5.5. Identify the fragment

Consume the remaining string starting with the leading "#" and to the end of the string.
Percent-encode all characters present from the ucschar list.
This is the fragment.
TODO: Handling of special characters "?" and "/"

6. Scheme-Specific Processing

TODO Apply limited scheme-specific rules. Reference [RFC4395]

6.1. http

(NOTE: Taken directly from [RFC3987]) For compatibility with existing deployed HTTP infrastructure, the following special case applies for schemes "http" and "https" and IRIs whose origin has a document charset other than one which is UCS-based (e.g., UTF-8 or UTF-16). In such a case, the "query" component of an IRI is mapped into a URI by using the document charset rather than UTF-8 as the binary representation before pct-encoding. This mapping is not applied for any other scheme or component.
Reference: Section 3.5 and 7.2 of [RFC3987].

6.2. javascript

TODO reference? http://tools.ietf.org/html/draft-hoehrmann-javascript-scheme-03

6.3. mailto

TODO reference [RFC6068]

6.4. data

TODO reference [RFC2397]

7. IRI Canonicalization

To follow.
[NOTE: Call out Special Case here or earlier? The U+005C should be either percent-encoded or converted to U+002F. Of course, folks who

want to name such files will want to use the escaped form of \ in order for their site to work in other browsers.

7.1. Producing a valid URI from an IRI

For each character which is not allowed anywhere in a valid URI, apply the following steps.
Reference: Section 3.5 of [RFC3987].

*Convert the IRI to the UTF-8 encoding, i.e., convert the character to a sequence of one or more octets using UTF-8 [RFC3629].
TODO: What about they query and fragment components? Should the

*For each IRI component, percent-encode each UTF-8 octet representing each character that is not allowed in the same URI component. In general this will include the set of characters specified by ucschar of [RFC3987].

8. Security Considerations

To follow.

9. IANA Considerations

This document has no actions for the IANA.

10. Acknowledgements

Many thanks to Mykyta Yevstifeyev, Addison Phillips, Mark Davis, Anne van Kesteren, Adam Barth, Martin Duerst, and Julian Reschke for their feedback.

11. References

11.1. Informative References

[2]	Barth, A, "How Browsers Process URLs", Internet-Draft draft-abarth-url-01, April 2011.
[3]	Hickson, I., "HTML Living Standard: URLs", WHATWG ?, 2011.
[4]	Fielding, R., "Change proposal for ISSUE-56", July 2010.
[5]	W3C, "URIs, URLs, and URNs: Clarifications and Recommendations 1.0", W3C Note 21 September 2001, September 2001.
[6]	Masinter, L., "HTML CHANGE PROPOSAL; change definition of URL to normative reference to IRIBIS", February 2010.
[7]	Davis, M., "Revision of UBA for improved display of URL/ IRIs", May 2011.
[8]	Davis, M. and M. Duerst, "URLs", April 2003.

11.2. Normative References

[RFC3987]	Duerst, M, Suignard, M and L Masinter, "Internationalized Resource Identifiers (IRIs)", Internet-Draft draft-ietf-iri-3987bis-05, March 2011.
[RFC3986]	Berners-Lee, T, Fielding, R and L Masinter, "Uniform Resource Identifier (URI)", Internet-Standard rfc3986, January 2005.
[RFC3629]	Yergeau, F, "UTF-8, a transformation format of ISO 10646", Internet-Standard rfc3629, November 2003.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Author's Address

Chris Weber Weber Casaba Security 16625 Redmond Wa, Suite M348 Redmond, WA 98052 USA EMail: chris@lookout.net