Guidelines and Registration Procedures for new URI/IRI Schemes 4395bis

Tony Hansen
November 9, 2010

4395bis IRI Registration

- draft-hansen-iri-4395bis-irireg-00
- draft-iri-4395bis-irireg-00

- 2 major issues raised on the list

Issue A

Bjorn Hoehrmann (Oct 5)

- "Previously, those who wish to describe resource identifiers that are useful as IRIs were encouraged to define the corresponding URI syntax, and note that the IRI usage follows the rules and transformations defined in [3987]. This document changes that advice to encourage explicit definition of the scheme and allowable syntax elements within the larger character repertoire of IRIs, as defined by [3987bis]."

- I am concerned that this would further draw a distinction between the characters that occur literally in an identifier and characters that are percent-encoded. I am not entirely sure in fact how to read RFC 3987 on this (it starts out saying it's just like URIs, except that there are more unreserved characters, but then excludes private use code points from the set of unreserved characters).

- Let's say I make a scheme where the scheme-specific part can only be "ö". Since "ö" is an unreserved character, I might be inclined to say def = "example:" %x00F6; but that would not work as "example:%c3%b6" is essentially defined as equivalent to "example:ö".
- The definition would have to account for a level of indirection at some point to remove percent-encoding, so I'd think you cannot quite distinguish between defining an URI scheme and an IRI scheme, so far the only difference could be in percent-encoded private use characters.
- I'd rather remove that difference, and am not sure what the actual change there would be.

Thoughts?

- Let's say I make a scheme where the scheme-specific part can only be "ö". Since "ö" is an unreserved character, I might be inclined to say def = "example:" %x00F6; but that would not work as "example:%c3%b6" is essentially defined as equivalent to "example:ö".
- The definition would have to account for a level of indirection at some point to remove percent-encoding, so I'd think you cannot quite distinguish between defining an URI scheme and an IRI scheme, so far the only difference could be in percent-encoded private use characters.
- I'd rather remove that difference, and am not sure what the actual change there would be.

Issue B (datatracker #49)

- A common confusion is that people think the fragment identifier is a scheme-specific, it's common for proposed registrations to define the fragment as part of the scheme, and it is unfortunately common that fragment identifiers are in fact treated as data, like "javascript:open('#example')" or "data:,#example" in implementations.
- However, fragment identifiers are part of the generic framework, the scheme-specific part ends where the fragment begins.
- I think 4395bis should discuss this problem in some detail.

Thoughts?

- A common confusion is that people think the fragment identifier is a scheme-specific, it's common for proposed registrations to define the fragment as part of the scheme, and it is unfortunately common that fragment identifiers are in fact treated as data, like "javascript:open('#example')" or "data:,#example" in implementations.
- However, fragment identifiers are part of the generic framework, the scheme-specific part ends where the fragment begins.
- I think 4395bis should discuss this problem in some detail.