

## Ruby 1.8 - Feature #1013

### Handling of [ and ] in URIs

01/15/2009 05:28 AM - tobias (Tobias Gruetzmacher)

<b>Status:</b>	Rejected
<b>Priority:</b>	Normal
<b>Assignee:</b>	akira (akira yamada)
<b>Target version:</b>	
<b>Description</b>	
=begin	
This is a little bit related to the old Tracker entry <a href="http://rubyforge.org/tracker/?group_id=426&amp;atid=1698&amp;func=detail&amp;aid=1952">http://rubyforge.org/tracker/?group_id=426&amp;atid=1698&amp;func=detail&amp;aid=1952</a>	
The problem surfaced for me again just today, while working with WWW::Mechanize parsing a private website (sorry, no public test case) using some Drupal image gallery software. It produces URIs like <a href="http://localhost/images/file[1].jpg">http://localhost/images/file[1].jpg</a> without escaping the [ and ].	
I know this is against the RFC (RFC 3986 states in section 3.2.2. Host: "This is the only place where square bracket characters are allowed in the URI syntax."), but I would be glad if Ruby was a little bit more liberal (in the old spirit of "Be liberal with what you accept and strict with what you produce") and accept these malformed URIs. Bonus points if it also corrected them...	
I had posted a "fix" to the old tracker item, but while it is a working workaround, it doesn't look like a valid solution to me anymore.	
This bug also applies to Ruby 1.9.	
=end	

### History

#### #1 - 02/02/2009 12:35 PM - ko1 (Koichi Sasada)

- Assignee set to akira (akira yamada)

=begin

=end

#### #2 - 02/05/2009 10:13 AM - akira (akira yamada)

- Status changed from Open to Rejected

=begin

[Ruby 1.8]

"[" and "]" can not be in URI::PATTERN::UNRESERVED because the URI module is used as a syntax checker of URI like strings.

[Ruby 1.9]

It is basically same as above. But you can use your URI parser. Example:

```
require "uri"
=> true
parser = URI::Parser.new(:UNRESERVED=>URI::PATTERN::UNRESERVED+'[]')
=> #URI::Parser:0xa09f9d4
uri = parser.parse("http://exmaple.jp/[1].jpg")
=> #
uri.path
=> "[1].jpg"
URI.parse("http://exmaple.jp/[1].jpg")
URI::InvalidURIError: bad URI(is not URI?): http://exmaple.jp/[1].jpg
```

=end