Ruby - Bug #15933

OpenURI: Assign default charset for HTTPS as well as HTTP

06/17/2019 07:14 PM - gareth (Gareth Adams)

Status:	Closed		
Priority:	Normal		
Assignee:	akr (Akira Tanaka)		
Target version:			
ruby -v:		**Backport:**	2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN

Description

Using open-uri to load a document in the following circumstances:

- The Content-Type header is text/* and *doesn't* specify a charset, e.g. Content-Type: text/csv
- The document is loaded from an https:// URL

…will cause the resulting string to have ASCII-8BIT encoding.

As the documentation for OpenURI#charset mentions, RFC2616/3.7.1 says:

> When no explicit charset parameter is provided by the sender, media subtypes of the "text" type are defined to have a default charset value of "ISO-8859-1" when received via HTTP.

OpenURI takes this literally - only assigning ISO-8859-1 if @base_uri.scheme is *exactly* "http". This check was written 17 years ago in 2002 even before TLS 1.1 was defined, and well before HTTPS was common.

I believe this check should now also match the scheme "https". As RFC2818/2 says:

> Conceptually, HTTP/TLS is very simple. Simply use HTTP over TLS precisely as you would use HTTP over TCP

1. Is this a suitable change to make?

2. I have a patch to fix the functionality (attached). What else do I need to specify in terms of documentation/tests? I'm happy to put more work into this, but it's my first contribution to Ruby core and I'd like some pointers. I've read through https://bugs.ruby-lang.org/projects/ruby/wiki/HowToReport

Associated revisions

Revision 8f7884761e30c453287d73de6ea733d565635ebc - 07/15/2019 12:36 AM - akr (Akira Tanaka)

The default charset of text/* media type is UTF-8.

Thanks for the patch gareth (Gareth Adams). [Bug #15933]

Combines two small, but very related changes

1: Treat HTTPS the same as HTTP

Previously, OpenURI followed guidance in RFC2616/3.7.1:

> When no explicit charset parameter is provided by the sender, media
> subtypes of the "text" type are defined to have a default charset
> value of "ISO-8859-1" when received via HTTP.

However this RFC was written before TLS was established and OpenURI was never updated to treat HTTPS traffic the same way. So, HTTPS documents received a different default to HTTP documents.

This commit removes the scheme check so that all text/* documents processed by OpenURI are treated the same way.

In theory this processing gets applied to FTP URIs too, but there's no mechanism in OpenURI for FTP documents to have Content-Type metadata appended to them, so this ends up being a no-op.

2: Change default charset for text/* to UTF-8

Replaces the default ISO-8859-1 charset previously defined in RFC2616 (now obsoleted) with a UTF-8 charset as defined in RFC6838.

Fixes: https://bugs.ruby-lang.org/issues/15933

Revision 8f7884761e30c453287d73de6ea733d565635ebc - 07/15/2019 12:36 AM - akr (Akira Tanaka)

The default charset of text/* media type is UTF-8.

Thanks for the patch gareth (Gareth Adams). [Bug #15933]

Combines two small, but very related changes

1: Treat HTTPS the same as HTTP

Previously, OpenURI followed guidance in RFC2616/3.7.1:

> When no explicit charset parameter is provided by the sender, media subtypes of the "text" type are defined to have a default charset value of "ISO-8859-1" when received via HTTP.

However this RFC was written before TLS was established and OpenURI was never updated to treat HTTPS traffic the same way. So, HTTPS documents received a different default to HTTP documents.

This commit removes the scheme check so that all text/* documents processed by OpenURI are treated the same way.

In theory this processing gets applied to FTP URIs too, but there's no mechanism in OpenURI for FTP documents to have Content-Type metadata appended to them, so this ends up being a no-op.

2: Change default charset for text/* to UTF-8

Replaces the default ISO-8859-1 charset previously defined in RFC2616 (now obsoleted) with a UTF-8 charset as defined in RFC6838.

Fixes: https://bugs.ruby-lang.org/issues/15933

Revision 8f788476 - 07/15/2019 12:36 AM - akr (Akira Tanaka)

The default charset of text/* media type is UTF-8.

Thanks for the patch gareth (Gareth Adams). [Bug #15933]

Combines two small, but very related changes

1: Treat HTTPS the same as HTTP

Previously, OpenURI followed guidance in RFC2616/3.7.1:

> When no explicit charset parameter is provided by the sender, media subtypes of the "text" type are defined to have a default charset value of "ISO-8859-1" when received via HTTP.

However this RFC was written before TLS was established and OpenURI was never updated to treat HTTPS traffic the same way. So, HTTPS documents received a different default to HTTP documents.

This commit removes the scheme check so that all text/* documents processed by OpenURI are treated the same way.

In theory this processing gets applied to FTP URIs too, but there's no

mechanism in OpenURI for FTP documents to have Content-Type metadata
appended to them, so this ends up being a no-op.

2: Change default charset for text/* to UTF-8

Replaces the default ISO-8859-1 charset previously defined in RFC2616 (now
obsoleted) with a UTF-8 charset as defined in RFC6838.

Fixes: https://bugs.ruby-lang.org/issues/15933

History

#1 - 06/17/2019 08:40 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Assigned

- Assignee set to akr (Akira Tanaka)

I think this change makes sense and the patch is the simplest way to implement it.

#2 - 06/17/2019 11:44 PM - phluid61 (Matthew Kerwin)

A lot of those quoted specs are very, very old, and in some cases obsoleted by newer specs.

HTTP/1.1 Semantics and Content RFC7231/B:

> The default charset of ISO-8859-1 for text media types has been
> removed; the default is now whatever the media type definition says.

Text Media Types RFC6838/4.2.1:

> If a "charset" parameter is specified, it SHOULD be a required
> parameter, eliminating the options of specifying a default value. If
> there is a strong reason for the parameter to be optional despite
> this advice, each subtype MAY specify its own default value, or
> alternatively, it MAY specify that there is no default value.
> Finally, the "UTF-8" charset [RFC3629] SHOULD be selected as the
> default. See [RFC6657] for additional information on the use of
> "charset" parameters in conjunction with subtypes of text.
>
> Regardless of what approach is chosen, all new text/* registrations
> MUST clearly specify how the charset is determined; relying on the
> US-ASCII default defined in Section 4.1.2 of [RFC2046] is no longer
> permitted. If explanatory text is needed, this SHOULD be placed in
> the additional information section of the registration.

Most current text/csv spec RFC7111/5.1

> The "charset" parameter specifies the charset employed by the CSV
> content. In accordance with RFC 6657 [RFC6657], the charset
> parameter SHOULD be used, and if it is not present, UTF-8 SHOULD
> be assumed as the default (this implies that US-ASCII CSV will
> work, even when not specifying the "charset" parameter). Any
> charset defined by IANA for the "text" tree may be used in
> conjunction with the "charset" parameter.

So it seems if you're making a change, it should be: ignore the protocol, and default to UTF-8 for text/csv.

#3 - 06/17/2019 11:58 PM - phluid61 (Matthew Kerwin)

phluid61 (Matthew Kerwin) wrote:

> So it seems if you're making a change, it should be: ignore the protocol, and default to UTF-8 for text/csv.

Or rather: ignore the protocol; and consult the IANA registry to see what the individual text/... types have as their default, and use UTF-8 as a final fallback. Which is unpleasant.

#4 - 06/18/2019 08:13 AM - gareth (Gareth Adams)

Thanks Matthew,

I've now paid more attention to which RFCs are obsolete and which are still active.

phluid61 (Matthew Kerwin) wrote:

> > So it seems if you're making a change, it should be: ignore the protocol, and default to UTF-8 for text/csv.
>
> Or rather: ignore the protocol; and consult the IANA registry to see what the individual text/... types have as their default, and use UTF-8 as a final fallback. Which is unpleasant.

The IANA registry isn't in a machine readable format, and so even if it were acceptable to depend on a gem like mime-types-data as a curated source of these values (I realise stdlib can't depend on gems), that data isn't currently available.

Looking through the registry manually, most text subtypes make no mention of a charset (either because they predate RFC6838 or because its recommendation to make charset required wasn't enforced) or specify UTF-8 explicitly. Only 5 (by my reading) mention a required charset parameter that is different to UTF-8, and in my opinion none of these are incompatible with using UTF-8 as a default.

text/sgml: specifies US-ASCII, but references an obsoleted RFC [RFC1521] to justify that.
text/troff: specifies US-ASCII, but cites "this will be the default 'US-ASCII'" and this specification predates RFC6838 which changed the default to UTF-8.
text/uri-list: See below.
text/vnd.a: specifies UTF-8 only "if 8 bit bytes are encountered" US-ASCII otherwise.
text/vnd.si.uricatalogue [obsoleted by author request]: specifies US-ASCII always.

The uri-list registration states (as of 1999):

> Currently, URIs can be represented using US-ASCII. However, there
> are many non-standard URIs which use special character sets.
> Discussion of how to best achieve internationalization of URIs is
> underway. This registration will be updated with a discussion of the
> URI charsets once that discussion has concluded.

The registration was not updated, despite IRIs being defined in RFC3987 to use UTF-8 or the ASCII transformation Punycode in 2005.

It seems to me that changing the default to UTF-8 and extending the check to match "https" URIs is:

- Correct in all cases except for a minuscule number of edge cases
- Compatible in all of those other cases
- Overridable by defining exceptions inline (as opposed to using a dependency like mime-types-data) if anyone raises issues with this default

My suggestion that we could override it (e.g. with a Hash of subtype => default_charset) is just as a contingency. There's no need to at the moment, and since this hasn't needed to be changed in nearly 20 years I'm not worried that this is a volatile piece of code.

If there are no objections, I'll follow up with a replacement patch using this as a plan.

#5 - 06/18/2019 09:24 AM - phluid61 (Matthew Kerwin)

gareth (Gareth Adams) wrote:

> The IANA registry isn't in a machine readable format, and so even if it were acceptable to depend on a gem like mime-types-data as a curated source of these values (I realise stdlib can't depend on gems), that data isn't currently available.

The entire registry is available as XML and each individual registry is available as (ironically) text/csv; e.g. https://www.iana.org/assignments/media-types/text.csv

That said, I agree in principle with pretty much everything else you've said.

> It seems to me that changing the default to UTF-8 and extending the check to match "https" URIs is:
>
> - Correct in all cases except for a minuscule number of edge cases
> - Compatible in all of those other cases
> - Overridable by defining exceptions inline (as opposed to using a dependency like mime-types-data) if anyone raises issues with this default

I would suggest ignoring the scheme altogether. Like:

```
diff a/lib/open-uri.rb b/lib/open-uri.rb
--- a/lib/open-uri.rb
+++ b/lib/open-uri.rb
@@ -552,7 +552,6 @@ def charset
       elsif block_given?
         yield
-       elsif type && %r{\Atext/} =~ type &&
-            @base_uri && /\Ahttp\z/i =~ @base_uri.scheme
-         "iso-8859-1" # RFC2616 3.7.1
+       elsif type && %r{\Atext/} =~ type
+         "utf-8" # RFC6838 4.2.1
       else
         nil
```

Cheers

#6 - 06/18/2019 12:02 PM - gareth (Gareth Adams)

phluid61 (Matthew Kerwin) wrote:

> The entire registry is available as XML and each individual registry is available as (ironically) text/csv; e.g. https://www.iana.org/assignments/media-types/text.csv

Sorry, I wasn't clear. Yes the registry itself is in XML/CSV – that's what the mime_types_data gem uses to build its dataset – but it doesn't include "default charset parameter" or even details of which parameters are required, so that has to be manually parsed out of the RFC/MIME registration. That's exactly what I did to get that shortlist of "conflicts" above.

I assumed your suggestion to "ignore the protocol; and consult the IANA registry" meant "at runtime" and I was clarifying that there was no way for that to be possible. If you didn't mean that (which was probably the more obvious way to look at it) then there's no problem at all.

> I would suggest ignoring the scheme altogether

I'm happy to do this, the only other case that's handled in this file is an ftp:// URI and the FTP parsing here doesn't have any way to extract metadata from the transferred file. Specifically it doesn't perform any file-extention-to-mime-type mapping, doesn't parse a content type, and so FTP URLs can't hit this branch of the code.

Thanks

#7 - 06/19/2019 12:59 PM - gareth (Gareth Adams)

- File ruby-changes.patch added

Updated patch attached

#8 - 06/27/2019 05:43 PM - gareth (Gareth Adams)

- File ruby-changes-combined.patch added

HowToContribute suggests that I can ping this ticket if it looks like it's been missed - there was some good discussion for a couple of days and then nothing since.

It's a small (+4 -5) change if anyone can weigh in.

The previous patch had it in two separate tiny commits but in case it's better, this patch is combined into one.

#9 - 07/15/2019 12:38 AM - akr (Akira Tanaka)

- Status changed from Assigned to Closed

Applied in changeset git|8f7884761e30c453287d73de6ea733d565635ebc.

The default charset of text/* media type is UTF-8.

Thanks for the patch gareth (Gareth Adams). [Bug #15933]

Combines two small, but very related changes

1: Treat HTTPS the same as HTTP

Previously, OpenURI followed guidance in RFC2616/3.7.1:

> When no explicit charset parameter is provided by the sender, media
> subtypes of the "text" type are defined to have a default charset
> value of "ISO-8859-1" when received via HTTP.

However this RFC was written before TLS was established and OpenURI was never updated to treat HTTPS traffic the same way. So, HTTPS documents received a different default to HTTP documents.

This commit removes the scheme check so that all text/* documents processed by OpenURI are treated the same way.

In theory this processing gets applied to FTP URIs too, but there's no mechanism in OpenURI for FTP documents to have Content-Type metadata appended to them, so this ends up being a no-op.

2: Change default charset for text/* to UTF-8

Replaces the default ISO-8859-1 charset previously defined in RFC2616 (now obsoleted) with a UTF-8 charset as defined in RFC6838.

Fixes: https://bugs.ruby-lang.org/issues/15933

Files

ruby-changes.patch	1.21 KB	06/17/2019	gareth (Gareth Adams)
ruby-changes.patch	3.05 KB	06/19/2019	gareth (Gareth Adams)
ruby-changes-combined.patch	2.24 KB	06/27/2019	gareth (Gareth Adams)