

## Redmine - Defect #24616

### Should not replace all invalid utf8 characters (e.g in mail)

2016-12-15 02:13 - Pavel Rosický

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	I18n	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	3.4.0	<b>Affected version:</b>	
<b>Resolution:</b>	Fixed		
<b>Description</b>			
<p>Hello,</p> <p>I've an email, that is encoded in utf8, but it contains an invalid character. In this case, redmine converts the content to us-ascii and then to utf8. This step will replace non-ascii compatible chars to "?". Why?</p> <p>1) Failure: MailHandlerTest#test_invalid_utf8 [/test/unit/mail_handler_test.rb:548]: Expected: "Здравствуйте?" Actual: "?????????????"</p> <p>I changed Redmine::CodesetUtil.replace_invalid_utf8(str) and Redmine::CodesetUtil.to_utf8(str, encoding)</p> <pre>str = str.encode("US-ASCII", :invalid =&gt; :replace, :undef =&gt; :replace, :replace =&gt; '?').encode("UTF-8")</pre> <p>to</p> <pre>str = str.encode("UTF-8", :invalid =&gt; :replace, :undef =&gt; :replace, :replace =&gt; '?')</pre> <p>all tests are passing with this change.</p> <p>Redmine version 3.3.1.stable Ruby version 2.1.5-p273 (2014-11-13) [x64-mingw32] Rails version 4.2.7.1 Environment production Database adapter MySQL2</p> <p>SCM: Git 2.10.1 Filesystem</p> <p>Redmine plugins: no plugin installed</p>			

#### Associated revisions

##### Revision 16271 - 2017-01-28 06:43 - Toshi MARUYAMA

add more tests (#24616)

##### Revision 16272 - 2017-01-28 06:43 - Toshi MARUYAMA

unify duplicate codes (#24616)

##### Revision 16273 - 2017-01-28 07:24 - Toshi MARUYAMA

do not replace all invalid utf8 (#24616)

##### Revision 16274 - 2017-01-28 07:25 - Toshi MARUYAMA

additional test for mail by Pavel Rosický (#24616)

##### Revision 16275 - 2017-01-28 07:29 - Toshi MARUYAMA

svn propset svn:eol-style native test/fixtures/mail\_handler/invalid\_utf8.eml (#24616)

## History

---

### #1 - 2016-12-29 08:14 - Go MAEDA

- File defect-24616.diff added

- Target version set to 3.3.2

Looks good to me.

```
# valid UTF-8 string
text = "?????"
p text.valid_encoding? # => true

# making invalid UTF-8 string
text.force_encoding('ASCII-8BIT')
text[-1] = 0xff.chr
text.force_encoding("UTF-8")
p text.valid_encoding? # => false
p text # => "????\xE3\x81\xFF"

# Current code of Redmine
p text.encode("US-ASCII", :invalid => :replace, :undef => :replace, :replace => '?').encode("UTF-8")
# => "?????"

# Fixed code by Pavel Rosický
p text.encode("UTF-8", :invalid => :replace, :undef => :replace, :replace => '?')
# => "?????"
```

### #2 - 2016-12-29 09:31 - Toshi MARUYAMA

- Target version deleted (3.3.2)

Did you run whole tests?

Especially this test.

[source:tags/3.3.1/test/unit/lib/redmine/codeset\\_util\\_test.rb](https://source.ruby-lang.org/ruby-devel/patches/3.3.1/test/unit/lib/redmine/codeset_util_test.rb)

### #3 - 2016-12-29 09:34 - Toshi MARUYAMA

Pavel Rosický wrote:

Hello,

I've an email, that is encoded in utf8, but it contains an invalid character. In this case, redmine converts the content to us-ascii and then to utf8. This step will replace non-ascii compatible chars to "?". Why?

You can see this function purpose.

[source:tags/3.3.1/test/unit/lib/redmine/codeset\\_util\\_test.rb#L68](https://source.ruby-lang.org/ruby-devel/patches/3.3.1/test/unit/lib/redmine/codeset_util_test.rb#L68)

### #4 - 2016-12-29 10:41 - Pavel Rosický

Thanks Toshi, I rechecked it again and all tests are passing.

[source:tags/3.3.1/test/unit/lib/redmine/codeset\\_util\\_test.rb#L68](https://source.ruby-lang.org/ruby-devel/patches/3.3.1/test/unit/lib/redmine/codeset_util_test.rb#L68)

In this case, my change has no effect on the result, because the string contains just one invalid utf-8 character.

```
s1.encode('us-ascii', :invalid => :replace, :undef => :replace, :replace => '?').encode('utf-8')
"Texte encod? en ISO-8859-1."

# patched
s1.encode('utf-8', :invalid => :replace, :undef => :replace, :replace => '?')
"Texte encod? en ISO-8859-1."
```

but a combination of valid and invalid utf-8 chars (non-ascii-compatible) will result both characters are stripped. Try out GO Media's example.

### #5 - 2017-01-28 05:19 - Toshi MARUYAMA

```
$ irb
1.9.3-p551 :001 > text = "????\xE3\x81\xFF"
=> "????\xE3\x81\xFF"
1.9.3-p551 :002 > text = text.encode("UTF-8", :invalid => :replace, :undef => :replace, :replace => '?')
=> "????\xE3\x81\xFF"
1.9.3-p551 :003 > text.valid_encoding?
=> false
```

```
$ irb
2.3.3 :001 > text = "\xE3\x81\xFF"
=> "\xE3\x81\xFF"
2.3.3 :002 > text = text.encode("UTF-8", :invalid => :replace, :undef => :replace, :replace => '?')
=> "\xE3\x81\xFF?"
2.3.3 :003 > text.valid_encoding?
=> true
```

#### #6 - 2017-01-28 07:07 - Toshi MARUYAMA

Pavel Rosický wrote:

Hello,

I've an email, that is encoded in utf8, but it contains an invalid character. In this case, redmine converts the content to us-ascii and then to utf8. This step will replace non-ascii compatible chars to "?". Why?

Because of Ruby 1.8.7 behavior compatibility.

[source:tags/2.6.9/lib/redmine/codeset\\_util.rb](https://source.ruby-lang.org/ruby-devel/2017/01/28/20170128070707-toshi-maruyama-1.html)

#### #7 - 2017-01-28 07:09 - Toshi MARUYAMA

- Subject changed from *encoding error if email contains an invalid utf8 character* to *Should not replace all invalid utf8 characters*

- Category changed from *Email receiving* to *I18n*

#### #8 - 2017-01-28 07:10 - Toshi MARUYAMA

- Subject changed from *Should not replace all invalid utf8 characters* to *Should not replace all invalid utf8 characters (e.g in mail)*

#### #9 - 2017-01-28 08:56 - Toshi MARUYAMA

- Status changed from *New* to *Closed*

- Target version set to *3.4.0*

- Resolution set to *Fixed*

I have committed [r16273](#) to pass on Ruby 1.9.3.

I don't want to change behavior on stable.

#### Files

<a href="#">invalid_utf8_test.patch</a>	1.49 KB	2016-12-15	Pavel Rosický
<a href="#">defect-24616.diff</a>	1.73 KB	2016-12-29	Go MAEDA