

## Ruby - Feature #4774

### User Friendly Handling of "Encoding::ConverterNotFoundError"

05/25/2011 01:57 AM - lazaridis.com (Lazaridis Ilias)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Target version:</b>		
<b>Description</b>		
Based on my own (user) experience with issue <a href="#">[#4738]</a> (which was solved), I suggest the following enhancement.		
=== 1st Step - Minimum ===		
The basic action would be to extend the error message "Encoding::ConverterNotFoundError" like this:		
"		
Converter XXXX to XXXX not found.		
More information on: <a href="http://ruby-lang.org/encoding-converter.html">http://ruby-lang.org/encoding-converter.html</a>		
"		
The document would explain why this error happened, and what the user can do now.		
Example:		
please run "ruby --encode-sanity" and send the output to ...		
Search in " <a href="http://redmine.ruby-lang.org">http://redmine.ruby-lang.org</a> " for "converter XXX to XXX"		
(something like that, that give a process to follow)		
A developer locates the relevant missing file, e.g.: "ibm737-tbl.rb", and place it into the sources for the next release.		
The existent installation of the user has still the problem - he has to wait for the next release.		
=== 2nd Step - Preliminary Encoders ===		
The user would have the possibility to drop-in a translation table file. The file "ibm737-tbl.rb" is published on the "encoding-converter" page, the user gets it and drops it into its local installation, e.g.:		
ruby192/lib/enc/trans/preliminary/		
The exception (Encoding::ConverterNotFoundError) tries to resolve the missing translation table with a lookup to a preliminary/*.rb table (which has lower speed, but it works).		
With the next release, the file would become redundant, because it was compiled in. A message could inform the user about it.		
=== Alternative - By Compilation ===		
If the compilation of the translation tables is unavoidable by design, then the site		
<a href="http://ruby-lang.org/encoding-converter.html">http://ruby-lang.org/encoding-converter.html</a>		
could point to the relevant compiled *.so files		
In this case, the missing *.so file would be generated out of "ibm737-tbl.rb", and provided for download (for the major OS, Win, Mac etc.)		
The files could be "stolen" from actual automated builds (don't know if currently available for ruby).		
=== gem based ===		
The final possible implementation (and possibly the most elegant one) would be, to have the preliminary encoding-translations within a gem.		

So, in this example, `ibm737-tbl.rb` would have been placed into the ruby sources, and the actual "trans" gem would have been updated and uploaded (with native versions for the major OS's).

The user would update his local installation by:

`chcp 850 => change codepage to 850 or something else, thus gem works`  
`gem install encoders => installs the encoding-translation which was just updated with ibm737-tbl.rb`  
`chcp 737 => change back to original codepage`

This way the user can continue with his install ruby immediately, and the sources are updated

(please note: I'm not familiar with the ruby sources. I've invested some hours to write this down, thus a core developer with the relevant domain knowledge can verify and possibly implement this.)

## History

### #1 - 05/27/2011 01:01 AM - naruse (Yui NARUSE)

The basic action would be to extend the error message "Encoding::ConverterNotFoundError" like this:

In current situation, users can do nothing, so such message is useless.  
Or such message is associated with the rdoc of the method which raises Encoding::ConverterNotFoundError.

A developer locates the relevant missing file, e.g.: "`ibm737-tbl.rb`", and place it into the sources for the next release.

We don't recommend such hack even if people may do it.

The existent installation of the user has still the problem - he has to wait for the next release.

He should report it before 1.9.2 is released.  
Especially IBM737, it is unintended missing file and should be reported.

The final possible implementation (and possibly the most elegant one) would be, to have the preliminary encoding-translations within a gem.

It is sometimes discussed, but not implemented yet.  
We consider it is possible but not easy.

### #2 - 05/28/2011 08:47 AM - lazaridis.com (Lazaridis Ilias)

=begin  
Yui NARUSE wrote:

The basic action would be to extend the error message "Encoding::ConverterNotFoundError" like this:

In current situation, users can do nothing, so such message is useless.

This is not exactly true.

- I did something (filed issue [#4738](#)), and my problem was solved.
- even if the user can do nothing: he should be informed

Or such message is associated with the rdoc of the method which raises Encoding::ConverterNotFoundError.

A new user does not know about rdoc.

Possibly he could be pointed to an issue here.

A developer locates the relevant missing file, e.g.: "`ibm737-tbl.rb`", and place it into the sources for the next release.

We don't recommend such hack even if people may do it.

This is what the developer has done in my case: [#4738](#)

The existent installation of the user has still the problem - he has to wait for the next release.

He should report it before 1.9.2 is released.

Nobody knows when a user hits on a "ConverterNotFoundError".

Especially IBM737, it is unintended missing file and should be reported.

I've reported it already, and the file was added to the repo: [#4738](#)

But I invest some time, thus ruby becomes better.

A new ruby user should be informed about this problem.

Imagine this scenario:

- Someone downloads and installs ruby, python and perl on his windows laptop.
- he starts to look around a little, and installs some software for all the 3 systems
- the ruby installation hits on the "ConverterNotFoundError"

He continues with perl and python, and finally gets exited about python, or:

- He follows the link in the error messages, follows the instructions, the team reacts and adds the missing transcoder
- He gets exited about how well organized all this, and gets exited about ruby.

The ruby team cannot leave users alone with this. The current design has a weakness (none knows on which machine the error will happen).

So, filing an issue (or pointing to a webpage) with information, and pointing to this from the error message, this is the very first step.

The final possible implementation (and possibly the most elegant one) would be, to have the preliminary encoding-translations within a gem.

It is sometimes discussed, but not implemented yet.

We consider it is possible but not easy.

You are language developers, so things can be difficult.

An easy extension of the existent system could(!) be:

(I don't know the name, I call it "transcode-library" for now)

Status:

- The transcode-library has an include path, where it fetches the transcoder \*.so's (e.g. enc/trans)

Changes:

- The transcode-library is modified, thus it can look for the \*.so's in multiple paths/locations
- The transcoder gem would add itself to the include path when installed (and remove itself, when deinstalled)
  - e.g. whilst calling "transcoder.add\_path"

(I can take the time to look deeper into this, if there's a possibility that it will be implemented)

=end

**#3 - 05/30/2011 12:47 AM - naruse (Yui NARUSE)**

Lazaridis Ilias wrote:

Yui NARUSE wrote:

The basic action would be to extend the error message "Encoding::ConverterNotFoundError" like this:

In current situation, users can do nothing, so such message is useless.

This is not exactly true.

- I did something (filed issue [#4738](#)), and my problem was solved.
- even if the user can do nothing: he should be informed

This problem is from missing converters even if Windows needs it.  
It is not good thing to say about general message of exception based on special case.

Or such message is associated with the rdoc of the method which raises Encoding::ConverterNotFoundError.

A new user does not know about rdoc.

Usually people found Encoding::ConverterNotFoundError when people use encoding conversion.

Possibly he could be pointed to an issue here.

A developer locates the relevant missing file, e.g.: "ibm737-tbl.rb", and place it into the sources for the next release.

We don't recommend such hack even if people may do it.

This is what the developer has done in my case: [#4738](#)

Commit to trunk is not "such hack".

The existent installation of the user has still the problem - he has to wait for the next release.

He should report it before 1.9.2 is released.

Nobody knows when a user hits on a "ConverterNotFoundError".

He should learn about it.

Especially IBM737, it is unintended missing file and should be reported.

I've reported it already, and the file was added to the repo: [#4738](#)

But I invest some time, thus ruby becommes better.

A new ruby user should be informed about this problem.

Imagine this scenario:

- Someone downloads and installs ruby, python and perl on his windows laptop.
- he starts to look around a little, and installs some software for all the 3 systems
- the ruby installation hits on the "ConverterNotFoundError"

This scenario doesn't happen other than people who uses Greek Windows with 1.9.2.  
On Ruby 1.9.3 there should be no environment.

Usually Encoding::ConverterNotFoundError is raised with exotic like emacs-mule.  
Your message is not suitable for such cases.

An easy extension of the existent system could(!) be:

(I don't know the name, I call it "transcode-library" for now)

It is "transcode".

Status:

- The transcode-library has an include path, where it fetches the transcoder \*.so's (e.g. enc/trans)

Changes:

- The transcode-library is modified, thus it can look for the \*.so's in multiple paths/locations
- The transcoder gem would add itself to the include path when installed (and remove itself, when deinstalled)
  - e.g. whilst calling "transcoder.add\_path"

(I can take the time to look deeper into this, if there's a possibility that it will be implemented)

Of course we know such changes (and some other related changes) make transcoders pluggable.  
I'm saying it is not easy.

#### #4 - 05/30/2011 06:09 AM - lazaridis.com (Lazaridis Ilias)

Yui NARUSE wrote:

Lazaridis Ilias wrote:

[...]

This problem is from missing converters even if Windows needs it.

[...]

Usually people found Encoding::ConverterNotFoundError when people use encoding conversion.

[...]

Especially IBM737, it is unintended missing file and should be reported.

[...]

This scenario doesn't happen other than people who uses Greek Windows with 1.9.2.

On Ruby 1.9.3 there should be no environment.

Usually Encoding::ConverterNotFoundError is raised with exotic like emacs-mule.

I start to understand that you want to tell me this:

- All transcode files for all *major* operating systems are integrated
- The "ibm737-tbl.rb" (relevant for Windows Greek) was accidentally not included to ruby 1.9
- From 1.9.3 on, no further transcode problems are expected for simple users

Is this so?

#### #5 - 06/03/2011 09:26 AM - naruse (Yui NARUSE)

Lazaridis Ilias wrote:

Yui NARUSE wrote:

Lazaridis Ilias wrote:

[...]

This problem is from missing converters even if Windows needs it.

[...]

Usually people found Encoding::ConverterNotFoundError when people use encoding conversion.

[...]

Especially IBM737, it is unintended missing file and should be reported.

[...]

This scenario doesn't happen other than people who uses Greek Windows with 1.9.2.

On Ruby 1.9.3 there should be no environment.

Usually Encoding::ConverterNotFoundError is raised with exotic like emacs-mule.

I start to understand that you want to tell me this:

- All transcode files for all *major* operating systems are integrated

Yes.

- The "ibm737-tbl.rb" (relevant for Windows Greek) was accidentally not included to ruby 1.9

Yes.

- From 1.9.3 on, no further transcode problems are expected for simple users

Yes, in the meaning of "run ruby and fail".

**#6 - 06/05/2011 06:15 PM - lazaridis.com (Lazaridis Ilias)**

Yui NARUSE wrote:

Lazaridis Ilias wrote:

Yui NARUSE wrote:  
[...]

I start to understand that you want to tell me this:

- All transcode files for all *major* operating systems are integrated

Yes.

- The "ibm737-tbl.rb" (relevant for Windows Greek) was accidentally not included to ruby 1.9

Yes.

- From 1.9.3 on, no further transcode problems are expected for simple users

Yes, in the meaning of "run ruby and fail".

I understand.

If it's so, then there is no need to do something more.

You can close/reject this issue (I cannot close it myself).

**#7 - 06/05/2011 07:17 PM - naruse (Yui NARUSE)**

- Status changed from Open to Closed

**#8 - 07/13/2011 05:50 PM - duerst (Martin Dürst)**

Yui NARUSE wrote:

Lazaridis Ilias wrote:

- All transcode files for all *major* operating systems are integrated

Yes.

Just for the record, this may not be true. Windows-1258, used for Vietnamese, is not supported (yet). The reason is that when converting it to (NFC) UTF-8, the conversion is not one-to-one by characters.

**#9 - 07/13/2011 07:46 PM - naruse (Yui NARUSE)**

Hmm, it can be implemented by the way I implemented UTF8-MAC.  
Will you implement it (on UTF8-MAC's way or some another way)?

**#10 - 07/14/2011 09:52 AM - duerst (Martin Dürst)**

Yui NARUSE wrote:

Hmm, it can be implemented by the way I implemented UTF8-MAC.

Will you implement it (on UTF8-MAC's way or some another way)?

Ah, okay, I'll have a look at UTF8-MAC. Regards, Martin.