Ruby master - Bug #18601
Invalid byte sequences in Big5 encodings
02/22/2022 10:15 PM - janosch-x (Janosch Müller)

Status: Open
Priority: Normal
Assignee: duerst (Martin Dürst)
Target version: ruby -v: any
Backport: 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN

Description
I encoded all unicode codepoints in all encodings:

```ruby
full_string = ((0..0xD7FF).to_a + (0xE000..0x10FFFF).to_a).pack('U*'); 1
uniq_encodings = Encoding.name_list - Encoding.aliases.keys - %w[locale external filesystem internal]
encoded_strings = uniq_encodings.map do |enc|
  full_string.encode(enc, invalid: :replace, undef: :replace, replace: '')
rescue => e
  puts e
end; 1
```

This prints about 10 "converter not found" errors, such as code converter not found (UTF-8 to UTF-7), but I guess this is expected.

Some of the converters seem to output invalid strings, though:

```ruby
encoded_strings.each do |str|
  str&.codepoints
rescue => e
  puts e
end; 1
```

This will print invalid byte sequence in {Big5HKSCS,Big5-UAO,CP950,CP951}.

Looking for example at the generated CP950 string, 8031 of its 25342 characters are invalid, spread across 2017 distinct ranges in the string. The invalid characters' codepoints are all in the range of 0x81..0xFE.

Is this a bug?

I would expect String#encode with invalid: :replace, undef: :replace not to create invalid byte sequences, but maybe I am misunderstanding these encodings and this is an unavoidable issue?

CC duerst (Martin Dürst)

History
#1 - 02/23/2022 07:59 AM - duerst (Martin Dürst)
  - Assignee set to duerst (Martin Dürst)

I'll try to take a closer look at this, but it will take a few days, sorry. Please ping me again if you don't hear back within a week or two.