# Redmine - Defect #29442

## Vendor-defined characters in ISO-2022-JP email subject break issue's subject

2018-08-27 13:17 - Go MAEDA

| | | | |
|---|---|---|---|
| **Status:** | Closed | **Start date:** | |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Go MAEDA | **% Done:** | 0% |
| **Category:** | Email receiving | **Estimated time:** | 0.00 hour |
| **Target version:** | 4.0.0 | | |
| **Resolution:** | Fixed | **Affected version:** | |

**Description**

When you create an issue via email, the subject field in the issue will be broken if the subject line in the email is encoded with ISO-2022-JP and contains vendor-defined characters such as "①".

Suppose a user is trying to create an issue via email with the subject "□□□□□□□" and it is encoded with ISO-2022-JP. Actually, the subject field of the issue will be **"$B-!(B $B4]?t;z%F%9%H(B"**.

This is because the email's subject contains the character "①". The CIRCLED DIGIT ONE character, U+2460 in Unicode, is not defined in ISO-2022-JP. But it is defined in some vendor-extended variants like ISO-2022-JP-MS and widely used in the real world. Probably Ruby would not process "①" in ISO-2022-JP and raises Encoding::UndefinedConversionError.

The undefined character "①" should be replaced with the replacement character instead of breaking the whole subject. Please see ":undef" and ":replace" option of String#encode in http://ruby-doc.org/core-2.5.1/String.html#method-i-encode.

The attached patch is a test to detect the problem.

```
$ ruby test/unit/mail_handler_test.rb
    .
  (snip)
    .
Encoding conversion failed "\xAD\xA1" to UTF-8 in conversion from ISO-2022-JP to stateless-ISO-202
2-JP to EUC-JP to UTF-8
Encoding conversion failed "\xAD\xA1" to UTF-8 in conversion from ISO-2022-JP to stateless-ISO-202
2-JP to EUC-JP to UTF-8
Encoding conversion failed "\xAD\xA1" to UTF-8 in conversion from ISO-2022-JP to stateless-ISO-202
2-JP to EUC-JP to UTF-8
Encoding conversion failed "\xAD\xA1" to UTF-8 in conversion from ISO-2022-JP to stateless-ISO-202
2-JP to EUC-JP to UTF-8
F

Failure:
MailHandlerTest#test_add_issue_with_iso_2022_jp_ms_subject [test/unit/mail_handler_test.rb:753]:
Expected /□□□□□□□/ to match "\e$B-!\e(B \e$B4]?t;z%F%9%H\e(B".

bin/rails test test/unit/mail_handler_test.rb:742
```

**Related issues:**

| | |
|---|---|
| Blocked by Redmine - Feature #29443: Update mail gem (~> 2.7.1) | **Closed** |

### Associated revisions

**Revision 17596 - 2018-10-15 12:00 - Go MAEDA**

Ensure that vendor-defined characters in ISO-2022-JP email subject don't break issue's subject (#29442).

Patch by Go MAEDA.

## History

#### #1 - 2018-08-27 13:22 - Go MAEDA

*- Target version set to Candidate for next minor release*

It will be fixed if we update mail gem to 2.7.0. See https://github.com/mikel/mail/pull/978 for details.

#### #2 - 2018-08-27 13:54 - Go MAEDA

*- Blocked by Feature #29443: Update mail gem (~> 2.7.1) added*

#### #3 - 2018-10-15 12:01 - Go MAEDA

*- Subject changed from Vendor-defined characters in ISO-2022-JP break issue's subject to Vendor-defined characters in ISO-2022-JP email subject break issue's subject*

*- Status changed from New to Closed*

*- Assignee set to Go MAEDA*

*- Target version changed from Candidate for next minor release to 4.0.0*

*- Resolution set to Fixed*

This issue has been fixed by #29443.

## Files

| | | | |
|---|---|---|---|
| broken-subject@2x.png | 11.6 KB | 2018-08-27 | Go MAEDA |
| test_add_issue_with_iso_2022_jp_ms_subject.diff | 1.61 KB | 2018-08-27 | Go MAEDA |