Ruby master - Feature #15742
Add Date#jisx0301 support new Japanese era
04/01/2019 04:17 AM - kaishuu0123 (Koki Oyatsu)

Status: Closed
Priority: Normal
Assignee: 
Target version: 

Description

Reproduce process

```ruby
irb(main):002:0> require 'date'
=> true
irb(main):003:0> Date.new(2019, 5, 1).jisx0301
=> "H31.05.01"
```

• ruby version: trunk

Result of reproduce process

```ruby
irb(main):003:0> Date.new(2019, 5, 1).jisx0301
=> "H31.05.01"
```

Expected result and the reason why you expect

```ruby
irb(main):002:0> require 'date'
=> true
irb(main):003:0> Date.new(2019, 5, 1).jisx0301
=> "R01.05.01"
```

reason

• new era (Reiwa?) starts from 2019/05/01
• refs: https://github.com/ruby/ruby/blob/trunk/ext/date/date_core.c#L7049
• I can't be sure that initial letter is "R".

Related issues:
Related to Ruby master - Feature #15195: Deal with new Japanese era
Closed

Blocks Ruby master - Bug #15740: Backport Unicode Version 12.1.0 and other Ja...
Closed

Associated revisions
Revision 320c98b4 - 04/09/2019 01:27 AM - nobu (Nobuyoshi Nakada)
date: support for Reiwa, new Japanese era

[Feature #15742]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@67474 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 67474 - 04/09/2019 01:27 AM - nobu (Nobuyoshi Nakada)
date: support for Reiwa, new Japanese era

[Feature #15742]

Revision d6387f70 - 04/13/2019 03:14 PM - naruse (Yui NARUSE)
merge revision(s) 66745,67397,67398,67399,67400,67409,67410,67411,67412,67425,67426,67473,67474; [Backport #15742]

Bump version to date-2.0.0.

I forgot to change it when Ruby 2.6.0 was released.
date_parse.c: extract Japanese era initials

* expand tabs.

date_parse.c: removed 'r' which is not in JIS X 0301 yet

Added tests for end of Heisei
date: use del_hash to extract an element destructively

* expand tabs.

date_parse.c: renamed JAPANESE prefix as JISX0301
date_parse.c: name JISX0301_DEFAULT_ERA
date: make zone a substring to copy encoding and taintedness

* expand tabs.

date_core.c: [DOC] Heisei will be assumed if no-era [ci skip]
date: support for Reiwa, new Japanese era

[Feature #15742]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_6@67527 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 67527 - 04/13/2019 03:14 PM - naruse (Yui NARUSE)
merge revision(s) 66745, 67397, 67399, 67400, 67409, 67410, 67412, 67425, 67426, 67473, 67474: [Backport #15742]
Bump version to date-2.0.0.

I forgot to change it when Ruby 2.6.0 was released.
date_parse.c: extract Japanese era initials

* expand tabs.

date_parse.c: removed 'r' which is not in JIS X 0301 yet

Added tests for end of Heisei
date: use del_hash to extract an element destructively

* expand tabs.

date_parse.c: renamed JAPANESE prefix as JISX0301
date_parse.c: name JISX0301_DEFAULT_ERA
date: make zone a substring to copy encoding and taintedness

* expand tabs.

date_core.c: [DOC] Heisei will be assumed if no-era [ci skip]
date: support for Reiwa, new Japanese era

[Feature #15742]

History

#1 - 04/01/2019 04:21 AM - kaishuu0123 (Koki Oyatsu)
- Description updated

#2 - 04/01/2019 04:26 AM - mame (Yusuke Endoh)
- Status changed from Open to Feedback

Thanks, that's good to know. However, we should wait for the new revision of JIS X 0301. Could you let us know if the spec is actually revised?

#3 - 04/01/2019 04:26 AM - znz (Kazuhiro NISHIYAMA)
It seems Date.jisx0301 supports Japanese era too.

#4 - 04/01/2019 04:28 AM - duerst (Martin Dürst)
- Related to Feature #15195: Deal with new Japanese era added

#5 - 04/01/2019 04:28 AM - duerst (Martin Dürst)
- Blocks Bug #15740: Backport Unicode Version 12.1.0 and other Japanese new era issues to Ruby 2.6 added

#6 - 04/01/2019 04:38 AM - duerst (Martin Dürst)
kaishuu0123 (Koki Oyatsu) wrote:
- new era (Reiwa?) starts from 2019/05/01
- refs: https://github.com/ruby/ruby/blob/trunk/ext/date/date_core.c#L7049
- I can't be sure that initial letter is "R".

I'm also not 100% sure the new letter will be "R", but given that all major Romanization methods
https://en.wikipedia.org/wiki/Nihon-shiki_romanization) use 'R' and not 'L', and "ei" and "wa" are already established from Heisei and Showa, I'd bet
quite a bit that it's Reiwa and 'R', and not something else.

#7 - 04/01/2019 05:00 AM - nobu (Nobuyoshi Nakada)
znz (Kazuhiro NISHIYAMA) wrote:
- It seems Date.jisx0301 supports Japanese era too.
- It defaults to Heisei if no era initial letter is found.
- I guess it probably would be expected to the new era now, but it is an incompatibility of course.

#8 - 04/01/2019 05:23 AM - duerst (Martin Dürst)
nobu (Nobuyoshi Nakada) wrote:
- It defaults to Heisei if no era initial letter is found.
- I guess it probably would be expected to the new era now, but it is an incompatibility of course.

Yes, incompatibilities like this are unavoidable with a date system such as the Japanese era names.

#9 - 04/01/2019 05:38 AM - duerst (Martin Dürst)
mame (Yusuke Endoh) wrote:
- Thanks, that's good to know. However, we should wait for the new revision of JIS X 0301. Could you let us know if the spec is actually revised?

It may be that the spec is revised, but this usually takes quite some time. I'd personally suggest we move ahead so that this new functionality can be
used as soon as possible in Ruby.

#10 - 04/01/2019 06:09 AM - nobu (Nobuyoshi Nakada)
Tentative patch.
https://github.com/nobu/ruby/pull/new/feature/japanese-reiwa

#11 - 04/01/2019 06:31 AM - nobu (Nobuyoshi Nakada)
Considering Heisei is just 30years, omission of era can cause deadly confusions now and coming decades.
I think it should be an error, rather than assuming the latest era.

#12 - 04/01/2019 07:57 AM - kaishuu0123 (Koki Oyatsu)
Considering Heisei is just 30years, omission of era can cause deadly confusions now and coming decades.

(As a premise, difficult to discard current spec.)

I was surprised that handled the omission of era in Date.jisx0301.
Because I think that it is better not to incorporate features specific to a specific country.
Basically, I think Date::jisx0301 may be added by gem.

#13 - 04/01/2019 10:17 AM - nobu (Nobuyoshi Nakada)

kaishuu0123 (Koki Oyatsu) wrote:

I was surprised that handled the omission of era in Date::jisx0301.

More surprisingly, it is required by JIS X 0301; its definition is basically non-era, and the era may be preceded.

必要によっては、年の表示の前に元号を識別する記号を付けてもよい。

If necessary, the year display may be preceded by a symbol identifying the era.

(translated by google)

Because I think that it is better not to incorporate features specific to a specific country.

Basically, I think Date::jisx0301 may be added by gem.

Totally agree, but not only the particular methods, a kitchen-sink Date.parse also incorporates the feature.

#14 - 04/01/2019 10:33 AM - nobu (Nobuyoshi Nakada)

- Tracker changed from Misc to Feature

nobu (Nobuyoshi Nakada) wrote:

kaishuu0123 (Koki Oyatsu) wrote:

I was surprised that handled the omission of era in Date::jisx0301.

More surprisingly, it is required by JIS X 0301; its definition is basically non-era, and the era may be preceded.

必要によっては、年の表示の前に元号を識別する記号を付けてもよい。

If necessary, the year display may be preceded by a symbol identifying the era.

Note: JIS X 0301 states the era symbol is optional, but doesn't mention the default era in the case it is omitted.

At the best, it will be defined by the application(s), I guess, like announcers in ISO/IEC 2022.

#15 - 04/01/2019 10:55 AM - duerst (Martin Dürst)

nobu (Nobuyoshi Nakada) wrote:

Note: JIS X 0301 states the era symbol is optional, but doesn't mention the default era in the case it is omitted.

At the best, it will be defined by the application(s), I guess, like announcers in ISO/IEC 2022.

It's a sign that the committee responsible for JIS X 0301 wasn't able to agree, or didn't have any really good idea how it would work, or both. Essentially, it's similar to the Y2K problem, except it occurs more often.

#16 - 04/01/2019 11:26 AM - akr (Akira Tanaka)

nobu (Nobuyoshi Nakada) wrote:

Note: JIS X 0301 states the era symbol is optional, but doesn't mention the default era in the case it is omitted.

At the best, it will be defined by the application(s), I guess, like announcers in ISO/IEC 2022.


I think no-era representation is similar to truncation of ISO 8601:2000. "truncation" provides a way to omit higher order component by mutual agreement: describe the year 1985 as 85 for example. (See Wikipedia https://en.wikipedia.org/wiki/ISO_8601#Truncated_representations )

I think no-era representation should also be interpreted by a mutual agreement. In Ruby context, the agreement should be represented by the reference manual. Unfortunately, the manual is not clear. The manual of Date::jisx0301 describes the format as just "some typical JIS X 0301 formats"
Apart from that, "truncation" is removed at ISO 8601:2004. So, it is reasonable to consider no-era representation as legacy feature. (I don't know the future version of JIS X 0301, though.)

My idea:

- Document Date.jisx0301 interpret no-era representation as Heisei. This is appropriate because it is a legacy feature. And this is compatible as current version.
- Support "R" for Reiwa. I think the initial letter R is almost fixed and no-one lose with it.

#17 - 04/02/2019 02:52 AM - rrroybbbean (RRRoy BBBean)
Can this be handled like thread exceptions?

```ruby
Thread.abort_on_exception = true
```

I like this approach because it's dynamic and changes the behavior for the current runtime.

```ruby
Date.japanese_era_exception = true
```

Or perhaps something like Firefox about:config

```ruby
Runtime.thread.abort_on_exception = true
Runtime.date.japanese.era_exception = true
```

#18 - 04/02/2019 06:25 AM - nobu (Nobuyoshi Nakada)
akr (Akira Tanaka) wrote:

I think no-era representation is similar to truncation of ISO 8601:2000. "truncation" provides a way to omit higher order component by mutual agreement: describe the year 1985 as 85 for example. (See Wikipedia https://en.wikipedia.org/wiki/ISO_8601#Truncated_representations)

It reminds me the second parameter to Time.parse:

```ruby
We can change the date used to infer our missing elements by passing a second object that responds to #mon, #day and #year, such as Date, Time or DateTime.
Apart from that, "truncation" is removed at ISO 8601:2004.
```

Good to know it.

So, it is reasonable to consider no-era representation as legacy feature. (I don't know the future version of JIS X 0301, though.)

Maybe, we should await next JIS X 0301?

My idea:

- Document Date.jisx0301 interpret no-era representation as Heisei. This is appropriate because it is a legacy feature. And this is compatible as current version.
- Support "R" for Reiwa. I think the initial letter R is almost fixed and no-one lose with it.

Patches updated.

#19 - 04/02/2019 07:41 AM - akr (Akira Tanaka)
nobu (Nobuyoshi Nakada) wrote:

Maybe, we should await next JIS X 0301?

My idea:

- Document Date.jisx0301 interpret no-era representation as Heisei. This is appropriate because it is a legacy feature. And this is compatible as current version.
- Support "R" for Reiwa. I think the initial letter R is almost fixed and no-one lose with it.

Patches updated.

#19 - 04/02/2019 07:41 AM - akr (Akira Tanaka)
nobu (Nobuyoshi Nakada) wrote:

Maybe, we should await next JIS X 0301?

I think it is too late. JIS X 0301 update schedule is not shown.

However, we can wait vendors such as Microsoft. Microsoft also needs the single letter abbreviation of Reiwa.
I hope some guideline including the abbreviation will appear.

Patches updated.

You missed URL.
I found: https://github.com/nobu/ruby/tree/feature/japanese-reiwa

#20 - 04/08/2019 05:34 AM - akr (Akira Tanaka)

- OpenJDK also uses "R": https://bugs.openjdk.java.net/browse/JDK-8205432

#21 - 04/08/2019 11:26 PM - nobu (Nobuyoshi Nakada)

akr (Akira Tanaka) wrote:

Maybe, we should await next JIS X 0301?

I think it is too late.
JIS X 0301 update schedule is not shown.

As Japanese government has claimed that only one month is enough for the transition, the new standard will be issued in this month at latest :)

#22 - 04/09/2019 01:27 AM - nobu (Nobuyoshi Nakada)

- Status changed from Feedback to Closed

Applied in changeset trunk/r67474.

date: support for Reiwa, new Japanese era

[Feature #15742]