Matz hope to integrate "did_you_mean" gem to Ruby 2.3

We need to solve to following issues.

- integrate interception gem
- naming(better name for did_you_mean)
- bundled policy(standard library? bundled gem?)

Related issues:
Related to Ruby master - Feature #11032: Add a warning for misspelling "def i... Closed

Associated revisions
Revision 39fa5d0 - 09/08/2015 12:54 PM - yuki
gems/bundled_gems: Automatically install the did_you_mean gem as a bundled gem.
[Feature #11252]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@51795 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 51795 - 09/08/2015 12:54 PM - yuki
gems/bundled_gems: Automatically install the did_you_mean gem as a bundled gem.
[Feature #11252]

Revision 51795 - 09/08/2015 12:54 PM - yuki
gems/bundled_gems: Automatically install the did_you_mean gem as a bundled gem.
[Feature #11252]

Revision 51795 - 09/08/2015 12:54 PM - yuki
gems/bundled_gems: Automatically install the did_you_mean gem as a bundled gem.
[Feature #11252]

Revision 51795 - 09/08/2015 12:54 PM - yuki
gems/bundled_gems: Automatically install the did_you_mean gem as a bundled gem.
[Feature #11252]

Revision 51795 - 09/08/2015 12:54 PM - yuki
gems/bundled_gems: Automatically install the did_you_mean gem as a bundled gem.
[Feature #11252]

History
#1 - 06/12/2015 08:04 AM - hsbt (Hiroshi SHIBATA)
   - Assignee set to hsbt (Hiroshi SHIBATA)

#2 - 06/12/2015 08:12 AM - matz (Yukihiro Matsumoto)
   Approved. Go ahead and resolve issues related to merge.

   Matz.

#3 - 06/16/2015 12:03 PM - yuki24 (Yuki Nishijima)
   It's a great honor and privilege for me to see the did_you_mean gem become part of the language and I am really grateful to this movement.
I've created a branch that only works on 2.3, but has less dependencies: https://github.com/yuki24/did_you_mean/compare/master...ruby-2.3

integrate interception gem

interception gem isn't actually mandatory since the branch uses TracePoint API directly to capture a binding object.

naming(better name for did_you_mean)

The only name I have in mind is correctable, but I'm open to suggestions. The main reason why I want to use a different name and force people to use a new one is that the current version of did_you_mean knows about too many external things including JRuby, Rubinius and Rails and I don't want Ruby to bundle it.

bundled policy(standard library? bundled gem?)

I would like the gem to be a bundled gem to minimize the maintenance cost of it, but a couple of features(e.g. NoMethodError#receiver: #10881, detailed error message) have to be implemented before it is bundled.

Thank you again for considering adding the did_you_mean gem to the language and I would love to do whatever I can to make it happen!

#4 - 06/17/2015 08:49 AM - duerst (Martin Dürst)
- Related to Feature #11032: Add a warning for misspelling "def intialize" added

#5 - 08/22/2015 07:04 AM - yuki24 (Yuki Nishijima)
We've discussed this at the Ruby developers meeting on Thursday and made several decisions(but still subject to change).

- We are not going to rename the gem (we'll just use did_you_mean)
- It'll be a bundled gem (Yuki will be responsible for releasing a version that works on Ruby 2.3 without C extensions)
- There should be a way of disabling this feature entirely (--disable-gem did_you_mean or --disable-gemname)

- There should be a method on NameError that returns a qualified name:

```ruby
class User; end

error = begin
  User::DoesntExist
  rescue NameError => e
  e
end

error.name                #=> :Exist
error.qualified_name      #=> :"User::DoesntExist"
```

The last 2 features can be separate features. That said, since the last one is a blocker, it would be great if this could be done before actually bundling the did_you_mean gem.

#6 - 09/08/2015 12:55 PM - Anonymous
- Status changed from Open to Closed

Applied in changeset r51795.

gems/bundled_gems: Automatically install the did_you_mean gem as a bundled gem.
[Feature #11252]

#7 - 09/09/2015 07:24 AM - Hanmac (Hans Mackowiak)
shouldnt that be fixed to "1.0.0.beta2" because "1.0.0.beta1" got yanked?