Ruby master - Bug #12667

full_gem_path is incorrect for core modules

08/10/2016 11:33 PM - jeffreyc (Jeff C)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>hsbt (Hiroshi SHIBATA)</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>ruby -v:</td>
<td>ruby 2.3.1p112 (2016-04-26 revision 54768) [x86_64-darwin15]</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN</td>
</tr>
</tbody>
</table>

Description

#find_full_gem_path assumes all gems are in gems_dir. It also leverages full_name, which assumes the directory includes the version. Neither is correct for core modules that behave like gems (e.g., JSON), which are located in lib/ruby/2.3.0 instead of lib/ruby/gems/2.3.0/gems and are unversioned. This causes #full_gem_path to return an incorrect value:

```ruby
> spec = Gem::Specification.load('~/.rbenv/versions/2.3.1/lib/ruby/gems/2.3.0/specifications/default/json-1.8.3.gemspec')
=> #<Gem::Specification:0x3fc7f944ef20 json-1.8.3>
> spec.full_gem_path
=> "~/.rbenv/versions/2.3.1/lib/ruby/gems/2.3.0/gems/json-1.8.3"
```

In this case, #full_gem_path should return ~/.rbenv/versions/2.3.1/lib/ruby/2.3.0/json.

This incorrect value causes anything that depends upon #full_gem_path to find the gem to fail; e.g., `bundle show`:

```bash
% bundle show json
The gem json has been deleted. It was installed at:
~/.rbenv/versions/2.3.1/lib/ruby/gems/2.3.0/gems/json-1.8.3
```

History

**#1 - 03/21/2019 12:33 AM - hsbt (Hiroshi SHIBATA)**
- Assignee set to hsbt (Hiroshi SHIBATA)
- Status changed from Open to Assigned

**#2 - 03/23/2019 07:52 AM - hsbt (Hiroshi SHIBATA)**
See https://github.com/bundler/bundler/issues/6930#issuecomment-475848401

**#3 - 10/18/2019 12:56 AM - jeremyevans0 (Jeremy Evans)**
- Status changed from Assigned to Closed