

## Ruby master - Feature #578

### add method to disassemble Proc objects

09/21/2008 03:10 AM - rogerdpack (Roger Pack)

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	ko1 (Koichi Sasada)
<b>Target version:</b>	1.9.2
<b>Description</b> =begin Currently RubyVM::InstructionSequence.disassemble works only for methods.  I wasn't sure how to patch RubyVM::InstructionSequence.disassemble to work with proc objects AND methods, so added a new method RubyVM::InstructionSequence.disassemble_proc which works for proc objects, in the accompanying patch. Either way would work.  Hope this can be included before the feature freeze, if possible :) --R =end	

#### History

##### #1 - 09/22/2008 02:07 PM - ko1 (Koichi Sasada)

- Assignee set to ko1 (Koichi Sasada)

=begin

=end

##### #2 - 09/28/2008 04:36 PM - yugui (Yuki Sonoda)

- Target version set to 2.0.0

=begin

=end

##### #3 - 10/11/2008 01:05 AM - rogerdpack (Roger Pack)

=begin

Question: any chance of getting this [or something like it] included before feature freezing all the way?

Thanks all. Ruby rox :)

=end

##### #4 - 11/24/2008 03:08 AM - ko1 (Koichi Sasada)

- Priority changed from Normal to 3

=begin

I want to fix it with modifying ISeq#disasm to receive Proc object on 1.9.2. Sorry for late response.

=end

##### #5 - 11/26/2008 10:57 PM - rogerdpack (Roger Pack)

=begin

thanks for looking into that :)

--R

=end

**#6 - 04/09/2009 03:56 AM - rogerdpack (Roger Pack)**

```
=begin
Any progress on this? any way I can help?
Thanks!
-=r
=end
```

**#7 - 07/07/2009 10:43 AM - nobu (Nobuyoshi Nakada)**

```
=begin
Hi,
```

At Sun, 21 Sep 2008 03:02:44 +0900,  
Roger Pack wrote in [ruby-core:18762]:

```
I wasn't sure how to patch
RubyVM::InstructionSequence.disassemble
to work with proc objects AND methods, so added a new method
RubyVM::InstructionSequence.disassemble_proc
which works for proc objects, in the accompanying patch.
Either way would work.
```

It's pretty easy.

Index: iseq.c

```
-----
--- iseq.c (revision 23974)
+++ iseq.c (working copy)
@@ -1007,4 +1007,13 @@ iseq_s_disasm(VALUE klass, VALUE body)
}
}
```

- else {
- rb\_proc\_t \*proc;
- VALUE iseqval;
- GetProcPtr(body, proc);
- iseqval = proc->block.iseq->self;
- if (RUBY\_VM\_NORMAL\_ISEQ\_P(iseqval)) {
- ret = rb\_iseq\_disasm(iseqval);
- }

• }

return ret;

Nobu Nakada

```
=end
```

**#8 - 07/07/2009 10:50 AM - rocky (Rocky Bernstein)**

```
=begin
Another possibility would be to add an instruction-sequence method (iseq) into Proc. Off of the instruction sequence, one could use the methods like disasm or to_a.
```

The actual additional code (put in the right place) for this seems pretty small:

```
VALUE
proc_iseq(VALUE self)
{
rb_proc_t *proc;
rb_iseq_t *iseq;
VALUE rb_iseq;
GetProcPtr(self, proc);
iseq = proc->block.iseq;
if (!iseq) return Qnil;
rb_iseq = iseq_alloc_shared(rb_cISeq);
RDATA(rb_iseq)->data = iseq;
return rb_iseq;
}
```

```
rb_define_method(rb_cProc, "iseq", proc_iseq, 0);
```

```
Warning: the above may have flaws in it.  
=end
```

**#9 - 10/06/2009 08:55 PM - rogerdpack (Roger Pack)**

```
=begin  
Would it be possible for somebody to commit this? Do we need more approval?  
Thanks.  
-r  
=end
```

**#10 - 10/06/2009 10:42 PM - matz (Yukihiro Matsumoto)**

```
=begin  
Hi,  
  
In message "Re: [ruby-core:25962] [Feature #578] add method to disassemble Proc objects"  
on Tue, 6 Oct 2009 20:55:57 +0900, Roger Pack redmine@ruby-lang.org writes:
```

```
|Would it be possible for somebody to commit this? Do we need more approval?
```

```
Yea, a patch from Nobu in [ruby-core:24169] needs permission from  
Koichi.
```

```
matz.
```

```
=end
```

**#11 - 10/08/2009 01:06 AM - ko1 (Koichi Sasada)**

```
=begin  
Roger Pack wrote::
```

```
Yea, a patch from Nobu in [ruby-core:24169] needs permission from  
Koichi.
```

```
Ok. Koichi do you have any feedback on the patch?
```

```
no. go ahead.
```

```
--  
// SASADA Koichi at atdot dot net
```

```
=end
```

**#12 - 10/08/2009 01:27 AM - nobu (Nobuyoshi Nakada)**

```
- Status changed from Open to Closed  
- % Done changed from 0 to 100
```

```
=begin  
Applied in changeset r25258.  
=end
```

**#13 - 11/11/2009 12:53 AM - rogerdpack (Roger Pack)**

```
=begin  
The fact that we can get backtraces from the VM easily almost makes me wonder if the following API is possible:
```

```
Thread#backtrace_with_bindings  
=>  
[[file, line, method, [[:arg1_name, current_value], [:arg2_name, current_value]]
```

```
or  
file, line, method, binding # so you can lookup argument values if so desired
```

```
mostly inspired by http://barelyenough.org/blog/2005/04/ruby-backtraces/ -- not sure if it's actually necessary/high priority since you can get somewhat  
higher quality backtraces using ruby-debug already (http://github.com/rdp/backtracer)
```

```
Thoughts?
```

-r  
=end

**#14 - 12/19/2009 11:52 AM - marcandre (Marc-Andre Lafortune)**

- *Category set to core*
- *Status changed from Closed to Open*
- *Target version changed from 2.0.0 to 1.9.2*

=begin

I didn't check the code, but isn't it supposed to make the following work?

```
$ rubydev -e 'RubyVM::InstructionSequence.disassemble(Proc.new{40+2})'  
-e:1:in disassemble': wrong argument type proc (expected method) (TypeError)  
from -e:1:in'
```

=end

**#15 - 01/20/2010 05:14 AM - rogerdpack (Roger Pack)**

- *File iseq.diff added*

=begin

This diff appears to fix it, if somebody could take a look at it.

Thanks.

-r

=end

**#16 - 01/24/2010 12:05 PM - rogerdpack (Roger Pack)**

- *Status changed from Open to Closed*

=begin

Appears fixed now. Thanks!

r26363

=end

**Files**

---

add_disassemble_proc.diff	889 Bytes	09/21/2008	rogerdpack (Roger Pack)
iseq.diff	786 Bytes	01/20/2010	rogerdpack (Roger Pack)