

Ruby master - Feature #14230

Binding#source_location

12/25/2017 09:38 AM - mame (Yusuke Endoh)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
How about providing Binding#source_location?	
<pre>b = binding p binding.source_location #=> ["test.rb", 1]</pre>	
To fetch the information, people uses a dirty idiom: <code>eval("__FILE__, __LINE__", binding)</code> .	
<ul style="list-style-type: none">https://github.com/ruby/ruby/blob/trunk/lib/irb/workspace.rb#L111https://github.com/rspec-given/rspec-given/blob/19e9ef24c159f5a014a9f22ebb94161d6ac5af43/lib/given/module_methods.rb#L35https://github.com/quix/live_ast/blob/7adaabdbb99dbc649fef4dfd8a821d9f347cc9f2/lib/live_ast/common.rb#L37https://github.com/wesabe/robot-army/blob/29c50d3a5f2258c917fef821103fb0f107edb428/lib/robot-army/eval_builder.rb#L14	
I noticed this issue when changing the semantics of eval so that it does not inherit <code>__FILE__</code> from the binding. [#4352]	
<pre>diff --git a/lib/irb/workspace.rb b/lib/irb/workspace.rb index 9ce84b60fa..5216520f32 100644 --- a/lib/irb/workspace.rb +++ b/lib/irb/workspace.rb @@ -108,7 +108,7 @@ def filter_backtrace(bt) end def code_around_binding - file, pos = @binding.eval('__FILE__', __LINE__) + file, pos = @binding.source_location unless defined?(::SCRIPT_LINES__[file]) && lines = ::SCRIPT_LINES__[file] begin diff --git a/proc.c b/proc.c index 9dbc78ae4a..23c4d3a2d4 100644 --- a/proc.c +++ b/proc.c @@ -12,6 +12,7 @@ #include "eval_intern.h" #include "internal.h" #include "gc.h" +#include "vm_core.h" #include "iseq.h" /* Proc.new with no block will raise an exception in the future @@ -611,6 +612,24 @@ bind_receiver(VALUE bindval) return vm_block_self(&bind->block); } +/* + * call-seq: + * binding.source_location -> [String, Integer] + * + * Returns the Ruby source filename and line number of the binding object. + */ +static VALUE +bind_location(VALUE bindval) +{ + VALUE loc[2];</pre>	

```

+   const rb_binding_t *bind;
+   GetBindingPtr(bindval, bind);
+   loc[0] = pathobj_path(bind->pathobj);
+   loc[1] = INT2FIX(bind->first_lineno);
+
+   return rb_ary_new4(2, loc);
+}
+
+static VALUE
+cfunc_proc_new(VALUE klass, VALUE ifunc, int8_t is_lambda)
+{
+@@ -3223,5 +3242,6 @@ Init_Binding(void)
+   rb_define_method(rb_cBinding, "local_variable_set", bind_local_variable_set, 2);
+   rb_define_method(rb_cBinding, "local_variable_defined?", bind_local_variable_defined_p, 1);
+   rb_define_method(rb_cBinding, "receiver", bind_receiver, 0);
+   rb_define_method(rb_cBinding, "source_location", bind_location, 0);
+   rb_define_global_function("binding", rb_f_binding, 0);
+}

```

Associated revisions

Revision 571e48b7 - 12/26/2017 08:38 AM - mame (Yusuke Endoh)

proc.c (bind_location): Add Binding#source_location

Fixes #14230

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@61480 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 61480 - 12/26/2017 08:38 AM - mame (Yusuke Endoh)

proc.c (bind_location): Add Binding#source_location

Fixes #14230

Revision 61480 - 12/26/2017 08:38 AM - mame (Yusuke Endoh)

proc.c (bind_location): Add Binding#source_location

Fixes #14230

Revision 3095f4ec - 02/21/2018 05:06 PM - mame (Yusuke Endoh)

- NEWS: add NEWS entry about Binding#source_location [Feature #14230]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62517 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62517 - 02/21/2018 05:06 PM - mame (Yusuke Endoh)

- NEWS: add NEWS entry about Binding#source_location [Feature #14230]

Revision 62517 - 02/21/2018 05:06 PM - mame (Yusuke Endoh)

- NEWS: add NEWS entry about Binding#source_location [Feature #14230]

History

#1 - 12/25/2017 12:51 PM - shevegen (Robert A. Heiler)

I like introspection so I am usually in agreement with all ways to inspect ruby's behaviour.

+1

I also agree that eval-binding is not pretty.

#2 - 12/26/2017 08:17 AM - matz (Yukihiko Matsumoto)

Agreed.

Matz.

#3 - 12/26/2017 08:38 AM - mame (Yusuke Endoh)

- Status changed from Open to Closed

Applied in changeset [trunk|r61480](#).

proc.c (bind_location): Add Binding#source_location

Fixes [#14230](#)