Ruby trunk - Bug #13844

Toplevel returns should fire ensures

08/28/2017 08:20 PM - headius (Charles Nutter)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	2.4.1	**Backport:**	2.2: DONTNEED, 2.3: DONTNEED, 2.4: DONE

Description

In the following contexts, a return always fires the ensure that wraps it:

```
[] ~/projects/ruby $ ruby -e 'def foo; return; ensure; p :x; end; foo'
:x

[] ~/projects/ruby $ ruby -e 'def foo; 1.times { begin; return; ensure; p :x; end }; end; foo'
:x
```

However the new 2.4 support for toplevel returns does *not* fire ensures:

```
$ ruby -e 'begin; return; ensure; p :x; end'
<no output>
```

I believe this is inconsistent with how returns work everywhere else (both valid and invalid returns always fire ensure) and it should be changed to match.

Related issues:

Related to Ruby trunk - Bug #14061: Top-level return does not execute ensure ... **Closed**

Associated revisions

Revision d6347a46 - 09/01/2017 12:43 AM - nobu (Nobuyoshi Nakada)

compile.c: ensure after toplevel return

- compile.c (iseq_compile_each0): toplevel returns should fire ensures. [ruby-core:82492] [Bug #13844]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@59708 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 59708 - 09/01/2017 12:43 AM - nobu (Nobuyoshi Nakada)

compile.c: ensure after toplevel return

- compile.c (iseq_compile_each0): toplevel returns should fire ensures. [ruby-core:82492] [Bug #13844]

Revision 59708 - 09/01/2017 12:43 AM - nobu (Nobuyoshi Nakada)

compile.c: ensure after toplevel return

- compile.c (iseq_compile_each0): toplevel returns should fire ensures. [ruby-core:82492] [Bug #13844]

Revision 59708 - 09/01/2017 12:43 AM - nobu (Nobuyoshi Nakada)

compile.c: ensure after toplevel return

- compile.c (iseq_compile_each0): toplevel returns should fire ensures. [ruby-core:82492] [Bug #13844]

Revision 1069d3f1 - 09/10/2017 04:53 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 59708: [Backport #13844]

```
    compile.c: ensure after toplevel return

    * compile.c (iseq_compile_each0): toplevel returns should fire
      ensures.  [ruby-core:82492] [Bug #13844]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_4@59812 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 59812 - 09/10/2017 04:53 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 59708: [Backport #13844]

```
compile.c: ensure after toplevel return

* compile.c (iseq_compile_each0): toplevel returns should fire
  ensures.  [ruby-core:82492] [Bug #13844]
```

History

#1 - 08/28/2017 08:20 PM - headius (Charles Nutter)

See https://github.com/jruby/jruby/issues/4761 for our placeholder bug for this missing feature.

#2 - 08/29/2017 06:30 AM - shevegen (Robert A. Heiler)

This is indeed weird that the blocks are treated differently depending on whether they
are defined in a method or outside of it. I do not assume that this was intentional.

On a side note, interestingly, the last variant aka:

```
begin
  return
ensure
  p :x
end
```

Yields no output when put into a .rb file and run via "ruby", but in IRB I get "LocalJumpError: unexpected return".

#3 - 08/29/2017 06:25 PM - headius (Charles Nutter)

I do not assume it was intentional, but it was codified by tests added along with the toplevel return change. See test_syntax.rb in test_return_toplevel, where it tests that each of the following lines should exit silently:

```
return; raise
begin return; rescue SystemExit; exit false; end
begin return; ensure exit false; end
begin ensure return; end
begin raise; ensure; return; end
begin raise; rescue; return; end
return false; raise
return 1; raise
```

The first ensure line there is in my opinion incorrect; it *should* exit with failure.

The funny thing about this case is that the first ensure is *not* expected to fire, and the other two *are* expected to fire. I don't think the intent of toplevel return was to immediately exit the current script. I think the intent was to unroll the stack back to above the require or load that loaded this script, exactly like unrolling the stack to the method that called this one in a normal method ensure.

#4 - 08/29/2017 06:26 PM - headius (Charles Nutter)

Oh also...IRB is a very weird beast. The code you enter isn't really running at toplevel; it's running in an eval that's likely within a block or method scope (I haven't looked recently). As a result, things you'd expect to work at toplevel (like returns, now) don't behave the same way. This bug is only about returns at the true top level scope in a Ruby file or -e script.

#5 - 08/31/2017 02:07 AM - nobu (Nobuyoshi Nakada)

- Backport deleted (2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN)

- ruby -v deleted (ruby 2.4.1p111 (2017-03-22 revision 58053) [x86_64-darwin16])

- Tracker changed from Bug to Feature

The behavior was intentional.

#6 - 09/01/2017 12:34 AM - nobu (Nobuyoshi Nakada)

- Backport set to 2.2: DONTNEED, 2.3: DONTNEED, 2.4: REQUIRED

- ruby -v set to 2.4.1

- Tracker changed from Feature to Bug

#7 - 09/01/2017 12:43 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset trunk|r59708.

---

compile.c: ensure after toplevel return

- compile.c (iseq_compile_each0): toplevel returns should fire ensures. [ruby-core:82492] [Bug #13844]

#8 - 09/03/2017 09:12 AM - Eregon (Benoit Daloze)

Agreed, ensure should run in all cases escaping the scope, even more so when it lexically encloses the top-level return.

#9 - 09/10/2017 04:54 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.2: DONTNEED, 2.3: DONTNEED, 2.4: REQUIRED to 2.2: DONTNEED, 2.3: DONTNEED, 2.4: DONE

ruby_2_4 r59812 merged revision(s) 59708.

#10 - 10/27/2017 01:04 PM - nobu (Nobuyoshi Nakada)

- Related to Bug #14061: Top-level return does not execute ensure if the file is loaded pr required added