

Ruby master - Feature #16924

instance_eval equivalent for RubyVM::InstructionSequence

05/30/2020 03:49 PM - Wuerfel21 (Ada Gottensträter)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

Abstract

Proposal to expose the ability to evaluate precompiled ISEQs in non-toplevel contexts.

Background

A common pattern used for certain kinds of DSL is to instance_eval entire files, like this:

```
__dsl_object.instance_eval(File.read(__path), __path, 0)
```

This has certain advantages and disadvantages over other methods of executing DSL code (mainly that the global namespace is not being polluted by either the DSL itself or the code running in the DSL). I find it very useful.

Ruby exposes the RubyVM::InstructionSequence API, allowing caching of parsing/compilation steps to speed up execution. However, the resulting code can only be evaluated at the top level, disallowing its use with DSLs designed to be instance_eval'd.

Proposal

Add a RubyVM::InstructionSequence#eval_in_instance method that could be used to replace

```
__dsl_object.instance_eval(File.read(__path), __path, 0)
```

with

```
iseq = RubyVM::InstructionSequence.compile_file(__path)
# iseq could be cached here
iseq.eval_in_instance(__dsl_object)
```

Note that while instance_eval makes local variables available if the evaluated argument is a string, this new method would not (as the local variables can't be known at compile time). This discrepancy already exists between RubyVM::InstructionSequence#eval and all other standard library functions named eval, so I assume it's okay.

This new method could also replace some use cases of Kernel#eval (any where the passing of local variables is not required).

Implementation

From a cursory glance at the relevant code, this seems like a fairly simple feature to implement. Maybe an additional parameter to #compile is needed to change the ISEQ type, I'm not sure. If no one who is more experienced at working with Ruby internals wants to step up to implement this feature, I will try it myself.

Discussion

There is a rejected feature [#12093](#) about evaluating ISEQs with a binding (which is not possible since the compiler needs to know the local variables of a binding to compile for it, thus the request got rejected). There was some talk about evaling inside an instance in the comments, but I assume that was forgotten about due to the ticket being closed. Therefore I think opening this new ticket is justified.