

Ruby - Bug #20455

rb_errinfo() inconsistent with \$! in the caller Ruby code

04/25/2024 05:35 PM - Eregon (Benoit Daloze)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 3.4.0dev (2024-04-25T08:12:47Z master 64bd8e41df) [x86_64-linux]	Backport: 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN
Description <p>This (slightly modified for clarity) test in ruby/spec demonstrates the unexpected result:</p> <pre>describe "rb_errinfo" do def err \$! end it "is cleared when entering a C method" do begin raise StandardError rescue \$!.class.should == StandardError err.class.should == StandardError @s.rb_errinfo().should == nil end end end</pre> <p>Why does rb_errinfo() return nil there, when \$! is set in the caller (and \$! isn't per frame but per thread, as shown with err)?</p> <p>Is this bug?</p> <p>If not, what is the logic and reason for clearing \$! when calling a method defined in C?</p>		

History

#1 - 04/25/2024 07:37 PM - Eregon (Benoit Daloze)

@ko1 (Koichi Sasada) (and the code) pointed me to \$! is rb_ec_get_errinfo, which walks the stack to find the first rescue/ensure (does it stop at the first ensure if it's an ensure without an active exception?) and uses ec->errinfo if not, while rb_errinfo is just GET_EC()->errinfo.

I'm not sure what this means in terms of semantics, that they are the same but only if no rescue/ensure on the stack, and otherwise they are fully separate?

It sounds like very odd semantics.

BTW the header docs at <https://github.com/ruby/ruby/blob/6f4f360fc46269eaba753cafe557519677a45a11/include/ruby/internal/error.h#L77-L84> seems not quite true and rather misleading.