

Ruby 1.8 - Bug #3359

Date::Format::Bag performance improvement

05/28/2010 11:36 AM - kstephens (Kurt Stephens)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
ruby -v:	ruby 1.8.7 (2009-06-12 patchlevel 174) [x86_64-linux]

Description

=begin

Date::Format::Bag spends a lot of time in method_missing for a limited number of method selectors.

The hack below removed ~ 3% of time from a ActiveRecord/Rails app that parses and formats many Date objects.

```
class Date::Format::Bag
  def method_missing(sel, args) # , &block)
    sel = sel.to_s
    t = sel.dup
    set = t.chomp!('=')
    t = t.intern
    if set
      value = @elem[t] = args[0]
      expr = "def #{sel}(arg); @elem[#{t.inspect}] = arg; end"
    else
      value = @elem[t]
    end
    #
    # There appear to be no callers like:
    #
    # e.foo(something)
    #
    # Thus do not interpret any arguments as
    # the *rest parameter creates Arrays that are
    # never referenced.
    #
    # expr = "def #{sel}(*rest); @elem[#{t.inspect}]; end"
    expr = "def #{sel}(); @elem[#{t.inspect}]; end"
  end
  expr = "class #{self.class}; #{expr}; end;"
  # $stderr.puts " *** #{self.class}##{sel} => #{expr}"
  eval(expr)
  value
end
end
```

Might be better to just enumerate all the possible Hash slot getter/setters using a simple class macro, since Date::Format::Bag is not used by anything outside of Date.

=end

History

#1 - 05/28/2010 11:38 AM - kstephens (Kurt Stephens)

=begin

This version is a bit more succinct:

```
class Date::Format::Bag
  def method_missing(sel, *args) # , &block)
    sel = sel.to_s
    t = sel.dup
    set = t.chomp!('=')
    t = t.intern
```

```
#
# There appear to be no callers like:
#
#   e.foo(something)
#
# Thus do not interpret any arguments as
# the *rest parameter since it creates Arrays
# that are never referenced in:
#
#   def foo(*rest); @elem[:foo]; end
#
expr = <<"END"
```

```
class #{self.class}
def #{t}(); @elem[#{t.inspect}]; end
def #{t}=(arg); @elem[#{t.inspect}] = arg; end
end
END
# $stderr.puts " **** #{self.class}##{sel} =>\n#{expr}"
eval(expr)
```

```
  if set
    @elem[t] = args[0]
  else
    @elem[t]
  end
end
end
```

end

=end