Ruby - Feature #15618

Implement Enumerator::Yielder#to_proc

02/24/2019 03:55 PM - knu (Akinori MUSHA)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		

Description

When writing an Enumerator block, you often want to delegate iteration to another method like this:

```
enum = Enumerator.new { |y|
  Dir.glob("*.rb") { |file|
    File.open(file) { |f| f.each_line { |line| y << line } }
  }
}
```

I think this is such a common pattern, but the { |var| y << var } part looks redundant compared to a normal iterator method being able to delegate to another method as simply as follows:

```
  def each(&block)
    @children.each(&block)
  end
```

So, I propose adding #to_proc to Yielder so you can directly pass a yielder object to another method as a block argument.

```
enum = Enumerator.new { |y|
  Dir.glob("*.rb") { |file|
    File.open(file) { |f| f.each_line(&y) }
  }
}
```

Yielder is all about yielding, so I think it's pretty obvious what it means.

History

#1 - 02/24/2019 05:16 PM - shevegen (Robert A. Heiler)

I have no particularly strong pro or con opinion on the functionality itself.

I have only one comment about syntax, though. While { |line| y << line } }
may or may not be redundant (let's leave that open for the moment), and is
definitely longer than the other variant suggested, I think it is actually a
simpler-to-understand syntax compared to the .each_line(&y) variant.

#2 - 03/11/2019 06:53 AM - matz (Yukihiro Matsumoto)

Sounds reasonable.

Matz.

#3 - 03/11/2019 09:54 AM - knu (Akinori MUSHA)

- Status changed from Open to Closed

Merged with r67211.

Files

0001-Implement-Enumerator-Yielder-to_proc.patch	3.21 KB	02/24/2019	knu (Akinori MUSHA)