



”Logging that makes us
happier”

By Reid Morrison, Software Architect
“Building mission-critical systems with Ruby and Rails”



Semantic Logger

- ◊ Semantic Data
 - ◊ Not just text
 - ◊ Computer understandable
 - ◊ Don't need complex grep parsing
 - ◊ Query data logged in Elasticsearch or MongoDB
- ◊ Mission Critical Systems
 - ◊ High concurrency
 - ◊ High performance
 - ◊ Non-blocking
 - ◊ Measure everything
 - ◊ Duration is a first class citizen



Example

```
require 'semantic_logger'

# Set the global default log level
SemanticLogger.default_level = :trace

# Log to a file, and use the colored formatter
SemanticLogger.add_appender('dev.log',
  &SemanticLogger::Appender::Base.colored_formatter)

# Create an instance of a logger
# Add the application/class name to every log message
logger = SemanticLogger['MyClass']
```



Example, ...

```
# Informational logging  
logger.info("Calling Supplier")
```

```
# Error information  
logger.error(  
    "Oops external call failed",  
    result: :failed,  
    reason_code: -10  
)
```

Example, ...

```
# Set this thread's name for when multiple threads are all logging at the same time
```

```
Thread.current.name = "main"
```

```
# Debug information
```

```
results = [ 5, 7, 2, 10 ]
```

```
logger.debug { "A total of #{results.inject(0) {|sum, i| i+sum }} were processed" }
```

```
# New level for logging low level trace information such as data sent or received
```

```
raw_response = "<xml><user>jbloggs</user><lastname>Bloggs</lastname><firstname>Joe</firstname></xml>"
```

```
logger.trace { "Raw data received from Supplier: #{raw_response.inspect}" }
```



Benchmarking

◦ Why is it taking so long to run?

```
count = User.where('created_at <= ?', date).count
```

Benchmarking

○ Wrap any code with logging:

```
count = logger.benchmark_info('Counting users') do  
  User.where('created_at <= ?', date).count  
end
```

2014-11-18 18:12:40.392160 I [63599:70098266504160] (303.1ms) RailsConf -- Counting users

Tagged Logging

◦ Wrap any code with logging:

```
logger.tagged(tracking_number) do

  count = logger.benchmark_info('Counting users') do
    User.where('created_at <= ?', date).count
  end

end
```

Metrics

- Add metrics to any benchmark
 - New Relic, any dashboarding solution

```
count = logger.benchmark_info('Counting users',  
  metric: 'Custom/count_users/user_management') do  
  
  User.where('created_at <= ?', date).count  
end
```



Appenders

- ◊ Screen
- ◊ File
- ◊ MongoDB
- ◊ Syslog
 - ◊ local, remote, UDP and TCP



Appenders

- New Relic
 - Send Exceptions to New Relic
 - Incl. Free New Relic Account
- Splunk
- bugsnag
- Roll your own
- All at the same time!



Asynchronous Logging

- ◊ Logging Thread
- ◊ Formatting
- ◊ Writes to appenders, screen, syslog
- ◊ No buffering
 - ◊ Logs are near realtime for viewing
 - ◊ Multiple processes writing to the same file - not a mess with partial lines from different processes

Example, output

```
2014-06-24 19:54:37.369442 I [73269:70110807251960] MyClass -- Calling Supplier
2014-06-24 19:54:37.373202 E [73269:70110807251960] MyClass -- Oops external call failed --
{:result=>:failed, :reason_code=>-10}
2014-06-24 19:54:37.383027 D [73269:main] MyClass -- A total of 24 were processed
2014-06-24 19:54:37.437584 T [73269:main] MyClass -- Raw data received from Supplier: "<xml>
<user>jbloggs</user><lastname>Bloggs</lastname><firstname>Joe</firstname></xml>"
2014-06-24 19:54:38.193422 I [73269:main] (750.6ms) MyClass -- Called external interface
2014-06-24 19:54:38.202865 I [73269:main] [jbloggs] MyClass -- Hello World
2014-06-24 19:54:38.202881 D [73269:main] [jbloggs] MyClass -- More messages
```



Questions?

Semantic Logger:

http://reidmorrison.github.io/semantic_logger/

Reid Morrison

@reidmorrison

reidmo@gmail.com

www.linkedin.com/in/reidmorrison