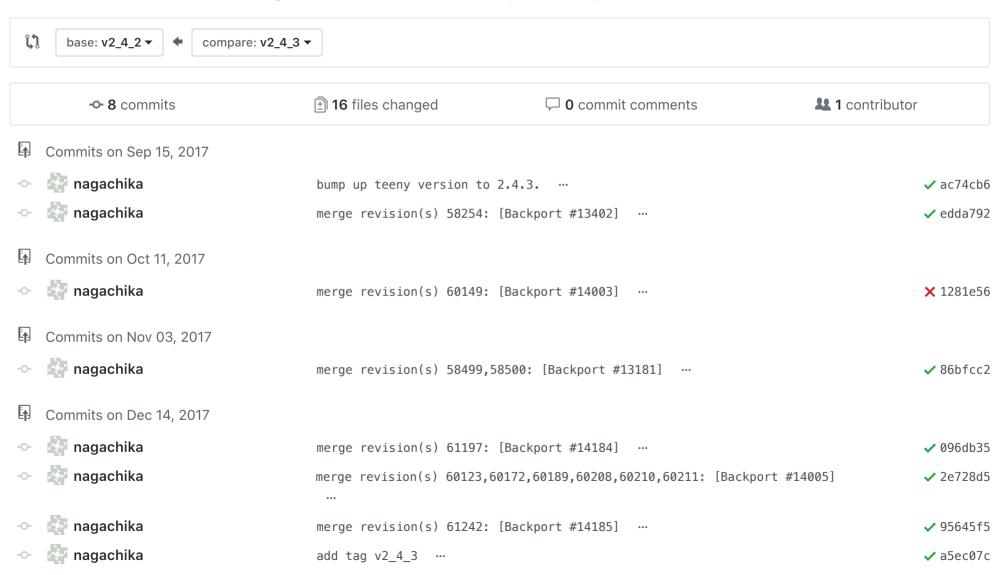# Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also **compare across forks**.

base: **v2_4_2** ← compare: **v2_4_3**

8 commits	16 files changed	0 commit comments	1 contributor

Commits on Sep 15, 2017

nagachika — bump up teeny version to 2.4.3. ⋯ — ✓ ac74cb6

nagachika — merge revision(s) 58254: [Backport #13402] ⋯ — ✓ edda792

Commits on Oct 11, 2017

nagachika — merge revision(s) 60149: [Backport #14003] ⋯ — ✗ 1281e56

Commits on Nov 03, 2017

nagachika — merge revision(s) 58499,58500: [Backport #13181] ⋯ — ✓ 86bfcc2

Commits on Dec 14, 2017

nagachika — merge revision(s) 61197: [Backport #14184] ⋯ — ✓ 096db35

nagachika — merge revision(s) 60123,60172,60189,60208,60210,60211: [Backport #14005] ⋯ — ✓ 2e728d5

nagachika — merge revision(s) 61242: [Backport #14185] ⋯ — ✓ 95645f5

nagachika — add tag v2_4_3 ⋯ — ✓ a5ec07c

Showing **16 changed files** with **415 additions** and **33 deletions**.

6 ■■■■□ `configure.in`

```
1403  1403        AS_IF([test "x$with_gmp" != xno],
1404  1404          [AC_CHECK_HEADERS(gmp.h)
1405  1405            AS_IF([test "x$ac_cv_header_gmp_h" != xno],
1406       -           AC_SEARCH_LIBS([__gmpz_init], [gmp]))])
      1406  +           AC_SEARCH_LIBS([__gmpz_init], [gmp],
      1407  +             [AC_DEFINE(HAVE_LIBGMP, 1)]))])
1407  1408
1408  1409      AC_ARG_WITH([jemalloc],
1409  1410        [AS_HELP_STRING([--with-jemalloc],[use jemalloc allocator])],
1410  1411        [with_jemalloc=$withval], [with_jemalloc=no])
1411  1412      AS_IF([test "x$with_jemalloc" = xyes],[
1412       -     AC_SEARCH_LIBS([malloc_conf], [jemalloc], [], [with_jemalloc=no])
      1413  +     AC_SEARCH_LIBS([malloc_conf], [jemalloc],
      1414  +       [AC_DEFINE(HAVE_LIBJEMALLOC, 1)], [with_jemalloc=no])
1413  1415      AC_CHECK_HEADER(jemalloc/jemalloc.h, [
1414  1416        AC_DEFINE(RUBY_ALTERNATIVE_MALLOC_HEADER, [<jemalloc/jemalloc.h>])
1415  1417      ])
```

10 ■■■■□ `lib/net/ftp.rb`

```
754  754          if localfile
755  755            if @resume
756  756              rest_offset = File.size?(localfile)
757       -          f = open(localfile, "a")
     757  +          f = File.open(localfile, "a")
758  758            else
759  759              rest_offset = nil
760       -          f = open(localfile, "w")
     760  +          f = File.open(localfile, "w")
761  761            end
```

```
762  762          elsif !block_given?
763  763            result = String.new
787  787          f = nil
788  788          result = nil
789  789          if localfile
790      -          f = open(localfile, "w")
     790  +          f = File.open(localfile, "w")
791  791          elsif !block_given?
792  792            result = String.new
793  793          end
833  833        else
834  834          rest_offset = nil
835  835        end
836      -      f = open(localfile)
     836  +      f = File.open(localfile)
837  837        begin
838  838          f.binmode
839  839          if rest_offset
852  852      # passing in the transmitted data one line at a time.
853  853      #
854  854      def puttextfile(localfile, remotefile = File.basename(localfile), &block) # :yield: line
855      -      f = open(localfile)
     855  +      f = File.open(localfile)
856  856        begin
857  857          storlines("STOR #{remotefile}", f, &block)
858  858        ensure
```

5 ▮▮▮▮▮▮ lib/rubygems.rb

```
10  10    require 'thread'
11  11
12  12    module Gem
13      -    VERSION = "2.6.13"
     13  +    VERSION = "2.6.14"
```

```
14    14        end
15    15
16    16      # Must be first since it unloads the prelude from 1.9.2
675   675
676   676          unless test_syck
677   677            begin
678   -                gem 'psych', '>= 1.2.1'
      678   +          gem 'psych', '>= 2.0.0'
679   679            rescue Gem::LoadError
680   680              # It's OK if the user does not have the psych gem installed.  We will
681   681              # attempt to require the stdlib version
699   699            end
700   700
701   701          require 'yaml'
      702   +        require 'rubygems/safe_yaml'
702   703
703   704          # If we're supposed to be using syck, then we may have to force
704   705          # activate it via the YAML::ENGINE API.
```

2 ■■■■■ lib/rubygems/config_file.rb

```
345   345          return {} unless filename and File.exist? filename
346   346
347   347          begin
348   -            content = YAML.load(File.read(filename))
      348   +          content = Gem::SafeYAML.load(File.read(filename))
349   349            unless content.kind_of? Hash
350   350              warn "Failed to load #{filename} because it doesn't contain valid YAML hash"
351   351              return {}
```

2 ■■■■■ lib/rubygems/package.rb

```
468   468
469   469          @checksums = gem.seek 'checksums.yaml.gz' do |entry|
```

```
470  470            Zlib::GzipReader.wrap entry do |gz_io|
471       -           YAML.load gz_io.read
     471  +           Gem::SafeYAML.safe_load gz_io.read
472  472          end
473  473        end
474  474      end
```

2 ▮▮▮▮▮ lib/rubygems/package/old.rb

```
101  101          header << line
102  102        end
103  103
104       -       YAML.load header
     104  +       Gem::SafeYAML.safe_load header
105  105      end
106  106
107  107      ##
```

48 ▮▮▮▮▮ lib/rubygems/safe_yaml.rb

```
...  ...      @@ -0,0 +1,48 @@
       1  + module Gem
       2  +
       3  +   ###
       4  +   # This module is used for safely loading YAML specs from a gem.  The
       5  +   # `safe_load` method defined on this module is specifically designed for
       6  +   # loading Gem specifications.  For loading other YAML safely, please see
       7  +   # # Psych.safe_load
       8  +
       9  +   module SafeYAML
      10  +     WHITELISTED_CLASSES = %w(
      11  +       Symbol
      12  +       Time
      13  +       Date
```

```
14  +         Gem::Dependency
15  +         Gem::Platform
16  +         Gem::Requirement
17  +         Gem::Specification
18  +         Gem::Version
19  +         Gem::Version::Requirement
20  +         YAML::Syck::DefaultKey
21  +         Syck::DefaultKey
22  +       )
23  +
24  +       WHITELISTED_SYMBOLS = %w(
25  +         development
26  +         runtime
27  +       )
28  +
29  +       if ::YAML.respond_to? :safe_load
30  +         def self.safe_load input
31  +           ::YAML.safe_load(input, WHITELISTED_CLASSES, WHITELISTED_SYMBOLS, true)
32  +         end
33  +
34  +         def self.load input
35  +           ::YAML.safe_load(input, [::Symbol])
36  +         end
37  +       else
38  +         warn "YAML safe loading is not available. Please upgrade psych to a version that supports safe loading (>= 2.0).
39  +         def self.safe_load input, *args
40  +           ::YAML.load input
41  +         end
42  +
43  +         def self.load input
44  +           ::YAML.load input
45  +         end
46  +       end
47  +     end
```

```
48    + end
```

2 ■■□□□ lib/rubygems/specification.rb

```
1101  1101            Gem.load_yaml
1102  1102
1103  1103            input = normalize_yaml_input input
1104        -         spec = YAML.load input
      1104  +         spec = Gem::SafeYAML.safe_load input
1105  1105
1106  1106            if spec && spec.class == FalseClass then
1107  1107              raise Gem::EndOfYAMLException
```

4 ■■■□ lib/webrick/httpserver.rb

```
267   267              k.sort!
268   268              k.reverse!
269   269              k.collect!{|path| Regexp.escape(path) }
270         -          @scanner = Regexp.new("^(" + k.join("|") +")(?=/|$)")
      270   +          @scanner = Regexp.new("\\A(" + k.join("|") +")(?=/|\\z)")
271   271            end
272   272
273   273            def normalize(dir)
274   274              ret = dir ? dir.dup : ""
275         -          ret.sub!(%r|/+$|, "")
      275   +          ret.sub!(%r|/+\z|, "")
276   276              ret
277   277            end
278   278          end
```

40 ■■■□ lib/webrick/server.rb

```
252   252            # the client socket.
```

```ruby
      def accept_client(svr)
        sock = nil
        begin
          sock = svr.accept
          sock.sync = true
          Utils::set_non_blocking(sock)
        rescue Errno::ECONNRESET, Errno::ECONNABORTED,
               Errno::EPROTO, Errno::EINVAL
        rescue StandardError => ex
          msg = "#{ex.class}: #{ex.message}\n\t#{ex.backtrace[0]}"
          @logger.error msg
        case sock = svr.to_io.accept_nonblock(exception: false)
        when :wait_readable
          nil
        else
          if svr.respond_to?(:start_immediately)
            sock = OpenSSL::SSL::SSLSocket.new(sock, ssl_context)
            sock.sync_close = true
            # we cannot do OpenSSL::SSL::SSLSocket#accept here because
            # a slow client can prevent us from accepting connections
            # from other clients
          end
          sock
        end
        return sock
      rescue Errno::ECONNRESET, Errno::ECONNABORTED,
             Errno::EPROTO, Errno::EINVAL
        nil
      rescue StandardError => ex
        msg = "#{ex.class}: #{ex.message}\n\t#{ex.backtrace[0]}"
        @logger.error msg
        nil
      end
```

```
268  276                     ##
269  277                         @logger.debug "accept: <address unknown>"
286  294                         raise
287  295                       end
288  296    +                 if sock.respond_to?(:sync_close=) && @config[:SSLStartImmediately]
     297    +                   WEBrick::Utils.timeout(@config[:RequestTimeout]) do
     298    +                     begin
     299    +                       sock.accept # OpenSSL::SSL::SSLSocket#accept
     300    +                     rescue Errno::ECONNRESET, Errno::ECONNABORTED,
     301    +                            Errno::EPROTO, Errno::EINVAL
     302    +                       Thread.exit
     303    +                     end
     304    +                   end
     305    +                 end
     306    +             call_callback(:AcceptCallback, sock)
289  307             block ? block.call(sock) : run(sock)
290  308           rescue Errno::ENOTCONN
291  309
```

20 ▪▪▪▪▪ parse.y

```
397  397    static int yylex(YYSTYPE*, struct parser_params*);
398  398
399  399    #ifndef RIPPER
     400  + static inline void
     401  + set_line_body(NODE *body, int line)
     402  + {
     403  +     if (!body) return;
     404  +     switch (nd_type(body)) {
     405  +       case NODE_RESCUE:
     406  +       case NODE_ENSURE:
     407  +         nd_set_line(body, line);
     408  +     }
     409  + }
```

```
410  +
400  411      #define yyparse ruby_yyparse
401  412
402  413      static NODE* node_newnode(struct parser_params *, enum node_type, VALUE, VALUE, VALUE);
2644 2655                      $$ = NEW_NIL();
2645 2656                  }
2646 2657                  else {
2647      -                     if (nd_type($3) == NODE_RESCUE ||
2648      -                         nd_type($3) == NODE_ENSURE)
2649      -                         nd_set_line($3, $<num>2);
     2658 +                     set_line_body($3, $<num>2);
2650 2659                      $$ = NEW_BEGIN($3);
2651 2660                  }
2652 2661                  nd_set_line($$, $<num>2);
2931 2940                  {
2932 2941                  /*****/
2933 2942                      $$ = NEW_CLASS($2, $5, $3);
     2943 +                     set_line_body($5, $<num>4);
2934 2944                      nd_set_line($$, $<num>4);
2935 2945                  /*%
2936 2946                      $$ = dispatch3(class, $2, $3, $5);
2950 2960                  {
2951 2961                  /*****/
2952 2962                      $$ = NEW_SCLASS($3, $6);
     2963 +                     set_line_body($6, nd_line($3));
2953 2964                      fixpos($$, $3);
2954 2965                  /*%
2955 2966                      $$ = dispatch2(sclass, $3, $6);
2973 2984                  {
2974 2985                  /*****/
2975 2986                      $$ = NEW_MODULE($2, $4);
     2987 +                     set_line_body($4, $<num>3);
2976 2988                      nd_set_line($$, $<num>3);
2977 2989                  /*%
```

```
2978   2990              $$ = dispatch2(module, $2, $4);
2997   3009              NODE *body = remove_begin($6);
2998   3010              reduce_nodes(&body);
2999   3011              $$ = NEW_DEFN($2, $5, body, METHOD_VISI_PRIVATE);
       3012  +           set_line_body(body, $<num>1);
3000   3013              nd_set_line($$, $<num>1);
3001   3014          /*%
3002   3015              $$ = dispatch3(def, $2, $5, $6);
3022   3035              NODE *body = remove_begin($8);
3023   3036              reduce_nodes(&body);
3024   3037              $$ = NEW_DEFS($2, $5, $7, body);
       3038  +           set_line_body(body, $<num>1);
3025   3039              nd_set_line($$, $<num>1);
3026   3040          /*%
3027   3041              $$ = dispatch5(defs, $2, $3, $5, $7, $8);
```

234 ▪▪▪▪▪ test/net/ftp/test_ftp.rb

```
   5      5    require "ostruct"
   6      6    require "stringio"
   7      7    require "tempfile"
          8  + require "tmpdir"
   8      9
   9     10    class FTPTest < Test::Unit::TestCase
  10     11      SERVER_NAME = "localhost"
2132   2133        end
2133   2134      end
2134   2135
       2136  +   def test_getbinaryfile_command_injection
       2137  +     skip "| is not allowed in filename on Windows" if windows?
       2138  +     [false, true].each do |resume|
       2139  +       commands = []
       2140  +       binary_data = (0..0xff).map {|i| i.chr}.join * 4 * 3
       2141  +       server = create_ftp_server { |sock|
```

```
2142  +          sock.print("220 (test_ftp).\r\n")
2143  +          commands.push(sock.gets)
2144  +          sock.print("331 Please specify the password.\r\n")
2145  +          commands.push(sock.gets)
2146  +          sock.print("230 Login successful.\r\n")
2147  +          commands.push(sock.gets)
2148  +          sock.print("200 Switching to Binary mode.\r\n")
2149  +          line = sock.gets
2150  +          commands.push(line)
2151  +          host, port = process_port_or_eprt(sock, line)
2152  +          commands.push(sock.gets)
2153  +          sock.print("150 Opening BINARY mode data connection for |echo hello (#{binary_data.size} bytes)\r\n")
2154  +          conn = TCPSocket.new(host, port)
2155  +          binary_data.scan(/.{1,1024}/nm) do |s|
2156  +            conn.print(s)
2157  +          end
2158  +          conn.shutdown(Socket::SHUT_WR)
2159  +          conn.read
2160  +          conn.close
2161  +          sock.print("226 Transfer complete.\r\n")
2162  +        }
2163  +        begin
2164  +          chdir_to_tmpdir do
2165  +            begin
2166  +              ftp = Net::FTP.new
2167  +              ftp.resume = resume
2168  +              ftp.read_timeout = 0.2
2169  +              ftp.connect(SERVER_ADDR, server.port)
2170  +              ftp.login
2171  +              assert_match(/\AUSER /, commands.shift)
2172  +              assert_match(/\APASS /, commands.shift)
2173  +              assert_equal("TYPE I\r\n", commands.shift)
2174  +              ftp.getbinaryfile("|echo hello")
2175  +              assert_equal(binary_data, File.binread("./|echo hello"))
```

```ruby
          assert_match(/\A(PORT|EPRT) /, commands.shift)
          assert_equal("RETR |echo hello\r\n", commands.shift)
          assert_equal(nil, commands.shift)
        ensure
          ftp.close if ftp
        end
      end
    ensure
      server.close
    end
  end

  def test_gettextfile_command_injection
    skip "| is not allowed in filename on Windows" if windows?
    commands = []
    text_data = <<EOF.gsub(/\n/, "\r\n")
foo
bar
baz
EOF
    server = create_ftp_server { |sock|
      sock.print("220 (test_ftp).\r\n")
      commands.push(sock.gets)
      sock.print("331 Please specify the password.\r\n")
      commands.push(sock.gets)
      sock.print("230 Login successful.\r\n")
      commands.push(sock.gets)
      sock.print("200 Switching to Binary mode.\r\n")
      commands.push(sock.gets)
      sock.print("200 Switching to ASCII mode.\r\n")
      line = sock.gets
      commands.push(line)
      host, port = process_port_or_eprt(sock, line)
```

```ruby
        commands.push(sock.gets)
        sock.print("150 Opening TEXT mode data connection for |echo hello (#{text_data.size} bytes)\r\n")
        conn = TCPSocket.new(host, port)
        text_data.each_line do |l|
          conn.print(l)
        end
        conn.shutdown(Socket::SHUT_WR)
        conn.read
        conn.close
        sock.print("226 Transfer complete.\r\n")
        commands.push(sock.gets)
        sock.print("200 Switching to Binary mode.\r\n")
      }
      begin
        chdir_to_tmpdir do
          begin
            ftp = Net::FTP.new
            ftp.connect(SERVER_ADDR, server.port)
            ftp.login
            assert_match(/\AUSER /, commands.shift)
            assert_match(/\APASS /, commands.shift)
            assert_equal("TYPE I\r\n", commands.shift)
            ftp.gettextfile("|echo hello")
            assert_equal(text_data.gsub(/\r\n/, "\n"),
                         File.binread("./|echo hello"))
            assert_equal("TYPE A\r\n", commands.shift)
            assert_match(/\A(PORT|EPRT) /, commands.shift)
            assert_equal("RETR |echo hello\r\n", commands.shift)
            assert_equal("TYPE I\r\n", commands.shift)
            assert_equal(nil, commands.shift)
          ensure
            ftp.close if ftp
          end
        end
```

```
2244  +      ensure
2245  +        server.close
2246  +      end
2247  +    end
2248  +
2249  +    def test_putbinaryfile_command_injection
2250  +      skip "| is not allowed in filename on Windows" if windows?
2251  +      commands = []
2252  +      binary_data = (0..0xff).map {|i| i.chr}.join * 4 * 3
2253  +      received_data = nil
2254  +      server = create_ftp_server { |sock|
2255  +        sock.print("220 (test_ftp).\r\n")
2256  +        commands.push(sock.gets)
2257  +        sock.print("331 Please specify the password.\r\n")
2258  +        commands.push(sock.gets)
2259  +        sock.print("230 Login successful.\r\n")
2260  +        commands.push(sock.gets)
2261  +        sock.print("200 Switching to Binary mode.\r\n")
2262  +        line = sock.gets
2263  +        commands.push(line)
2264  +        host, port = process_port_or_eprt(sock, line)
2265  +        commands.push(sock.gets)
2266  +        sock.print("150 Opening BINARY mode data connection for |echo hello (#{binary_data.size} bytes)\r\n")
2267  +        conn = TCPSocket.new(host, port)
2268  +        received_data = conn.read
2269  +        conn.close
2270  +        sock.print("226 Transfer complete.\r\n")
2271  +      }
2272  +      begin
2273  +        chdir_to_tmpdir do
2274  +          File.binwrite("./|echo hello", binary_data)
2275  +          begin
2276  +            ftp = Net::FTP.new
2277  +            ftp.read_timeout = 0.2
```

```ruby
          ftp.connect(SERVER_ADDR, server.port)
          ftp.login
          assert_match(/\AUSER /, commands.shift)
          assert_match(/\APASS /, commands.shift)
          assert_equal("TYPE I\r\n", commands.shift)
          ftp.putbinaryfile("|echo hello")
          assert_equal(binary_data, received_data)
          assert_match(/\A(PORT|EPRT) /, commands.shift)
          assert_equal("STOR |echo hello\r\n", commands.shift)
          assert_equal(nil, commands.shift)
        ensure
          ftp.close if ftp
        end
      end
    ensure
      server.close
    end
  end

  def test_puttextfile_command_injection
    skip "| is not allowed in filename on Windows" if windows?
    commands = []
    received_data = nil
    server = create_ftp_server { |sock|
      sock.print("220 (test_ftp).\r\n")
      commands.push(sock.gets)
      sock.print("331 Please specify the password.\r\n")
      commands.push(sock.gets)
      sock.print("230 Login successful.\r\n")
      commands.push(sock.gets)
      sock.print("200 Switching to Binary mode.\r\n")
      commands.push(sock.gets)
      sock.print("200 Switching to ASCII mode.\r\n")
      line = sock.gets
```

```
2312  +          commands.push(line)
2313  +          host, port = process_port_or_eprt(sock, line)
2314  +          commands.push(sock.gets)
2315  +          sock.print("150 Opening TEXT mode data connection for |echo hello\r\n")
2316  +          conn = TCPSocket.new(host, port)
2317  +          received_data = conn.read
2318  +          conn.close
2319  +          sock.print("226 Transfer complete.\r\n")
2320  +          commands.push(sock.gets)
2321  +          sock.print("200 Switching to Binary mode.\r\n")
2322  +        }
2323  +      begin
2324  +        chdir_to_tmpdir do
2325  +          File.open("|echo hello", "w") do |f|
2326  +            f.puts("foo")
2327  +            f.puts("bar")
2328  +            f.puts("baz")
2329  +          end
2330  +          begin
2331  +            ftp = Net::FTP.new
2332  +            ftp.connect(SERVER_ADDR, server.port)
2333  +            ftp.login
2334  +            assert_match(/\AUSER /, commands.shift)
2335  +            assert_match(/\APASS /, commands.shift)
2336  +            assert_equal("TYPE I\r\n", commands.shift)
2337  +            ftp.puttextfile("|echo hello")
2338  +            assert_equal(<<EOF.gsub(/\n/, "\r\n"), received_data)
2339  + foo
2340  + bar
2341  + baz
2342  + EOF
2343  +            assert_equal("TYPE A\r\n", commands.shift)
2344  +            assert_match(/\A(PORT|EPRT) /, commands.shift)
2345  +            assert_equal("STOR |echo hello\r\n", commands.shift)
```

```ruby
            assert_equal("TYPE I\r\n", commands.shift)
            assert_equal(nil, commands.shift)
          ensure
            ftp.close if ftp
          end
        end
      ensure
        server.close
      end
    end

    private

    def create_ftp_server(sleep_time = nil)
      end
      return data_server
    end

    def chdir_to_tmpdir
      Dir.mktmpdir do |dir|
        pwd = Dir.pwd
        Dir.chdir(dir)
        begin
          yield
        ensure
          Dir.chdir(pwd)
        end
      end
    end
  end
```

13 ■■■■■ test/ruby/test_parse.rb

```ruby
      end;
```

```
981   981          end
982   982
      983   +    def test_method_location_in_rescue
      984   +      bug = '[ruby-core:79388] [Bug #13181]'
      985   +      obj, line = Object.new, __LINE__+1
      986   +      def obj.location
      987   +        #
      988   +        raise
      989   +      rescue
      990   +        caller_locations(1, 1)[0]
      991   +      end
      992   +
      993   +      assert_equal(line, obj.location.lineno, bug)
      994   +    end
      995   +
983   996      =begin
984   997        def test_past_scope_variable
985   998          assert_warning(/past scope/) {catch {|tag| eval("BEGIN{throw tag}; tap {a = 1}; a")}}
```

25 ▇▇▇▇▇ test/webrick/test_httpserver.rb

```
415   415        }
416   416        assert_equal(0, requested, "Server responded to #{requested} requests after shutdown")
417   417      end
      418   +
      419   +  def test_cntrl_in_path
      420   +    log_ary = []
      421   +    access_log_ary = []
      422   +    config = {
      423   +      :Port => 0,
      424   +      :BindAddress => '127.0.0.1',
      425   +      :Logger => WEBrick::Log.new(log_ary, WEBrick::BasicLog::WARN),
      426   +      :AccessLog => [[access_log_ary, '']],
      427   +    }
```

```
428  +        s = WEBrick::HTTPServer.new(config)
429  +        s.mount('/foo', WEBrick::HTTPServlet::FileHandler, __FILE__)
430  +        th = Thread.new { s.start }
431  +        addr = s.listeners[0].addr
432  +
433  +        http = Net::HTTP.new(addr[3], addr[1])
434  +        req = Net::HTTP::Get.new('/notexist%0a/foo')
435  +        http.request(req) { |res| assert_equal('404', res.code) }
436  +        exp = %Q(ERROR `/notexist\\n/foo' not found.\n)
437  +        assert_equal 1, log_ary.size
438  +        assert log_ary[0].include?(exp)
439  +      ensure
440  +        s&.shutdown
441  +        th&.join
442  +      end
418  443        end
```

27 ███ test/webrick/test_ssl_server.rb

```
  2    2      require "webrick"
  3    3      require "webrick/ssl"
  4    4      require_relative "utils"
       5    + require 'timeout'
  5    6
  6    7      class TestWEBrickSSLServer < Test::Unit::TestCase
  7    8        class Echo < WEBrick::GenericServer
 37   38            io.close
 38   39          }
 39   40        end
       41    +
       42    +    def test_slow_connect
       43    +      poke = lambda do |io, msg|
       44    +        begin
       45    +          sock = OpenSSL::SSL::SSLSocket.new(io)
```

```ruby
        sock.connect
        sock.puts(msg)
        assert_equal "#{msg}\n", sock.gets, msg
      ensure
        sock&.close
        io.close
      end
    end
    config = {
      :SSLEnable => true,
      :SSLCertName => "/C=JP/O=www.ruby-lang.org/CN=Ruby",
    }
    Timeout.timeout(10) do
      TestWEBrick.start_server(Echo, config) do |server, addr, port, log|
        outer = TCPSocket.new(addr, port)
        inner = TCPSocket.new(addr, port)
        poke.call(inner, 'fast TLS negotiation')
        poke.call(outer, 'slow TLS negotiation')
      end
    end
  end
end
```

8 ◼◻◼◼◻ version.h

```c
@@ -1,9 +1,9 @@
- #define RUBY_VERSION "2.4.2"
- #define RUBY_RELEASE_DATE "2017-09-14"
- #define RUBY_PATCHLEVEL 198
+ #define RUBY_VERSION "2.4.3"
+ #define RUBY_RELEASE_DATE "2017-12-14"
+ #define RUBY_PATCHLEVEL 205

  #define RUBY_RELEASE_YEAR 2017
```

```diff
6     -    #define RUBY_RELEASE_MONTH 9
      6 +    #define RUBY_RELEASE_MONTH 12
7     7      #define RUBY_RELEASE_DAY 14
8     8
9     9      #include "ruby/version.h"
```

No commit comments for this range