

1 Ruby Tools

```
"lib/nio/tools.rb" 1a ≡
  # Common Utilities
  <License 1b>
  <Required Modules 1c>
  <definitions ?>
  <classes ?>
  module Nio
    <Nio classes 2a>
    module_function
    <Nio functions ?>
  end
  ◇

<License 1b> ≡
  # Copyright (C) 2003–2005, Javier Goizueta <javier@goizueta.info>
  #
  # This program is free software; you can redistribute it and/or
  # modify it under the terms of the GNU General Public License
  # as published by the Free Software Foundation; either version 2
  # of the License, or (at your option) any later version.
  ◇
```

Macro referenced in [1a](#).

```
<Required Modules 1c> ≡
  require 'rubygems'
  ◇
```

Macro referenced in [1a](#).

2 Tests

```
"test/test_tools.rb" 1d ≡

  #require File.dirname(__FILE__) + '/test_helper.rb'
  require 'test/unit'

  require 'nio/repdec'
  include Nio
  require 'yaml'

  <Tests definitions 2b>

  class TestTools < Test::Unit::TestCase

    def setup
      <Tests setup ?>
    end

    <Tests 2c>
  end
  ◇
```

3 State-Equivalent Classes

This mix-in module by Robert Klemme makes a class's equality-behaviour be based on object state (instance variables).

```

< Nio classes 2a > ≡
  module StateEquivalent
    def ==(obj); test_equal(obj); end
    def eql?(obj); test_equal(obj); end
    def ===(obj); test_equal(obj); end
    def hash
      h = 0
      self.instance_variables.each do |var|
        v = self.instance_eval var
        h ^= v.hash unless v.nil?
      end
      h
    end

    private
    def test_equal(obj)
      return false unless self.class == obj.class
      (self.instance_variables + obj.instance_variables).uniq.each do |var|
        v1 = self.instance_eval var
        v2 = obj.instance_eval var
        return false unless v1 == v2
      end
      true
    end
  end
  ◇

```

Macro referenced in [1a](#).

3.1 Tests

```

< Tests definitions 2b > ≡
  class SEclass
    include StateEquivalent
    def initialize (a,b)
      @a = a
      @b = b
    end
  end
  ◇

```

Macro referenced in [1d](#).

```

< Tests 2c > ≡
  def test_StateEquivalent
    x = SEclass.new(11,22)
    y = SEclass.new(11,22)
    z = SEclass.new(11,23)
    xx = x
    assert_equal(true,x==xx)
    assert_equal(true,x==y)
    assert_equal(false,x==z)
    assert_equal(x.hash,xx.hash)
    assert_equal(x.hash,y.hash)
    assert_equal(false,x.hash==z.hash)
  end
  ◇

```

Macro referenced in [1d](#).

4 Índices

4.1 Archivos

"lib/nio/tools.rb" Defined by [1a](#).

"test/test_tools.rb" Defined by [1d](#).

4.2 Fragmentos

⟨ License [1b](#) ⟩ Referenced in [1a](#).

⟨ Nio classes [2a](#) ⟩ Referenced in [1a](#).

⟨ Nio functions ? ⟩ Referenced in [1a](#).

⟨ Required Modules [1c](#) ⟩ Referenced in [1a](#).

⟨ Tests definitions [2b](#) ⟩ Referenced in [1d](#).

⟨ Tests setup ? ⟩ Referenced in [1d](#).

⟨ Tests [2c](#) ⟩ Referenced in [1d](#).

⟨ classes ? ⟩ Referenced in [1a](#).

⟨ definitions ? ⟩ Referenced in [1a](#).

4.3 Identificadores