Test Support

Version 5.0.2

Kathryn Gray

November 6, 2010

Contents

1 Using Check Forms

```
(require test-engine/racket-tests)
```

This module provides test forms for use in Racket programs, as well as parameters to configure the behavior of test reports.

Each check form may only occur at the top-level; results are collected and reported by the test function. Note that the check forms only register checks to be performed. The checks are actually run by the `test` function.

```
(check-expect test expected) → void?
  test : any/c
  expected : any/c
```

Accepts two value-producing expressions and structurally compares the resulting values.

It is an error to produce a function value or an inexact number.

```
(check-within test expected delta) → void?
  test : any/c
  expected : any/c
  delta : number?
```

Like `check-expect`, but with an extra expression that produces a number delta. Every number in the first expression must be within delta of the cooresponding number in the second expression.

It is an error to produce a function value.

```
(check-error test msg) → void?
  test : any/c
  msg : string?
(check-error test) → void?
  test : any/c
```

Checks that evaluating the first expression signals an error, where the error message matches the string, if it is present.

```
(check-member-of (test any/c) (expected any/c) ...)
```

Accepts at least two value-producing expressions. Structurally compares the first value to each value subsequent value specified.

It is an error to produce a function value.

```
(check-range (test number/c) (min number/c) (max number/c))
```

Accepts three number-producing expressions. Performs the following comparison: min <= test <= max.

```
(test) → void?
```

Runs all of the tests specified by check forms in the current module and reports the results. When using the gui module, the results are provided in a separate window, otherwise the results are printed to the current output port.

```
(test-format) → (any/c . -> . string?)
(test-format format) → void?
  format : (any/c . -> . string?)
```

A parameter that stores the formatting function for the values tested by the check forms.

```
(test-silence) → boolean?
(test-silence silence?) → void?
  silence? : any/c
```

A parameter that stores a boolean, defaults to #f, that can be used to suppress the printed summary from test.

```
(test-execute) → boolean?
(test-execute execute?) → void?
  execute? : any/c
```

A parameter that stores a boolean, defaults to #t, that can be used to suppress evaluation of test expressions.

2 GUI Interface

```
(require test-engine/racket-gui)
```

This module requires GRacket and produces an independent window when displaying test results. It provides the same bindings as `test-engine/racket-tests`.

3 Integrating languages with Test Engine

(To be written.)