Optimization Coach

Version 9.2.0.2

Vincent St-Amour <stamourv@racket-lang.org>

April 17, 2026

This package provides *optimization coaching* support to help you get the most of the Racket and Typed Racket optimizers.

The Optimization Coach DrRacket plugin can be used when editing a Typed Racket program in DrRacket. Clicking the **Optimization Coach** button runs the optimizer and reports the results. All performed optimizations are highlighted in green in the editor. In addition, the optimizer also reports cases where an optimization was close to happening, but was not ultimately safe to perform. These cases are highlighted in shades of red in the editor. The redder the highlight, the higher the potential for optimization in the highlighted region is.

Additional information can be accessed by right-clicking on the highlighted regions and picking the **Show Optimization Info** menu entry. A summary of the performed optimizations and advice on how to adjust code to make it more amenable to optimization is provided as appropriate, and can serve as a starting point for further optimization.

Optimization Coach is also available for other Racket languages through the **Show Optimization Coach** entry in the **View** menu. When running from unytped Racket languages, Optimization Coach does not report information about Typed Racket optimizations, and only reports information from the Racket optimizer.

You can exit the coach by clicking the **Close** button.

For more information about Optimization Coach's capabilities, see our OOPSLA 2012 paper. Note that there have been multiple extensions added since its publication.

1 Refining Recommendations with Profiling Information

Given profiling information about your program, Optimization Coach can tailor its recommendations to help you focus on the parts of your program that really matter.

```
(optimization-coach-profile
  #:use-errortrace? [u-e? #t]
  body ...)
```

To gather profiling information for use with Optimization Coach, wrap the portion of your program that you want to profile (typically an entry point to the program) with `optimization-coach-profile`.

When you next run your program, profiling information will be written to a file, ready to be used by Optimization Coach. The output filename is constructed by appending the `.profile` suffix to the program's filename.

By default, Optimization Coach uses errortrace for profiling (if errortrace-based profiling is available for your version of Racket). Errortrace-based profiling only profiles non-compiled files, or files compiled with errortrace annotations. You may need to delete your program's `compiled` directories before profiling. To enable errotrace support at the command-line:

```
racket -l errortrace -t file.rkt
```

To instead use the basic Racket profiler (using best-effort stack traces from the runtime system), set the `#:use-errortrace?` argument to `#f`.

Once you have gathered profiling information, you can feed it to Optimization Coach by specifying the profile file and clicking the **Refine** button. Optimization Coach will then reanalyze your program and produce new recommendations.

Compared to the pre-profiling recommendations, those new recommendations should be both more targeted and more aggressive. Post profiling, Optimization Coach only recommends changes to functions that had a significant impact on program performance according to profile data. These are the functions where your tuning efforts are likely best spent.

In addition, Optimization Coach's post-profiling recommendations are more aggressive. For example, it may recommend replacing convenient, high-level constructs—such as structs–with more performant but lower-level ones—such as vectors.

2 Verbose Mode

Optimization Coach provides a *verbose* mode, which is enabled by clicking the **Show More** button. In verbose mode, reports that would only be displayed for hot functions are displayed for all functions instead. No profiling information is required to enable verbose mode.