

OOPSLA 17 Artifact - Getting Started Guide

Fabian Muehlboeck, Cornell University
Ross Tate, Cornell University
(`{fabianm,ross}@cs.cornell.edu`)

Paper: Sound Gradual Typing is Nominally Alive and Well

1 Structure of the Artifact

1.1 Choices of Platform

Our compiler was developed on Windows, and it turns out that C# programs in the style of the benchmarks we used don't fare well on Mono or the Core CLR on Ubuntu. Hence, the best way to run all of our benchmarks is on Windows. We provide both a Virtual Machine image and standalone installation files and instructions for Windows, but both have the catch that one needs a valid Windows license (in case of the Virtual Machine, specifically for Windows 7 Professional SP1 x64). Thus, we also provide a Ubuntu-based Virtual Machine, and standalone installation files and instructions for Ubuntu. These later two options have the same functionality as their Windows counterparts except for the C# benchmarks.

NOTE: This file accompanies a smaller version of the artifact that excludes the virtual machine files. Please download the supplementary material archive from the ACM Digital Library.

1.2 Artifact Files

The files in this minimal artifact archive are organized as follows:

- **Getting Started.pdf**
This file.
- **Step By Step.pdf**
A guide for running and evaluating the artifact.
- **Ubuntu/**
A Folder containing the files needed for running the benchmark on an existing Ubuntu installation

- **Windows/**
A Folder containing the files needed for running the benchmark on an existing Windows installation

2 Windows Standalone installation

In order to set up a Windows installation (7 or higher) to run the benchmarks there, you need to:

- Install the .NET 4.6.2 Developer Pack (<https://www.microsoft.com/en-us/download/details.aspx?id=53321>)
- Install the Visual Studio C++ Build Tools 2015 (<http://landinghub.visualstudio.com/visual-cpp-build-tools>). Use a custom installation, the Windows 8 SDK is sufficient.
- Install Racket 6.2 (<http://download.racket-lang.org/racket-v6.2.html>)
- Install Python 3.6.2 (<https://www.python.org/ftp/python/3.6.2/python-3.6.2.exe>). In the installer, choose add to PATH, choose pip, tcl/tk, and the py launcher; precompile standard libraries and add to environment
- Add your Racket installation directory to the PATH environment variable
- Add the folder that contains vcvarsall.bat to the PATH environment variable. By default, that should be

`C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\`

(You should be able to find it by looking at the start menu items that start the VS 2015 [...] Tools Command Prompts)

- Copy the files from **Standalone/Windows** to somewhere on your machine.
- before running any individual benchmark, run `setup.bat` in the **BENCHMARKS** folder. `run_all_benchmarks.bat` runs it, too.

The artifact benchmark files include a folder called **csship**, which in turn contains the file `compile_cs.bat`. This script is used to compile the different C# programs - if it can compile C# programs (you can test that by copying it into one of the specific benchmark program folders that contain `.cs` files and running it there), the only thing left to do is to make sure that **Benchmark.exe** can run without complaining about missing the right version of the .NET framework.

2.1 Garbage Collector

The standalone files come with a precompiled version of the Boehm Garbage Collector (<http://hboehm.info/gc/index.html>). It may not work with a different version of Windows than the one we used. In this case, you will need to download

the garbage collector source and compile it according to your target's needs. Then, replace `gc.lib` in `BENCHMARKS/ngship`. If you are also planning to use debugging symbols via the `compile_c.bat` script, compile a separate version of the garbage collector for 32-bit mode and with debugging symbols and replace `gc_debugx86.lib` in `BENCHMARKS/ngship`, too.

3 Ubuntu Standalone installation

Required software:

- Mono (Ubuntu packages `mono-runtime` and `libmono-system-core4.0-cil`)
- Python 3 (preinstalled on Ubuntu) + PIP (Ubuntu package `python3-pip`) + Tkinter (Ubuntu package `python3-tk`) + matplotlib (`pip3 install matplotlib`)
- gcc (preinstalled on Ubuntu)
- Racket 6.2 (<http://download.racket-lang.org/racket-v6.2.html>)

Copy the files from `Standalone/Ubuntu` onto your machine somewhere. Before running any individual benchmark, run `raco link benchmark-util` in the `BENCHMARKS` folder. `run_all_benchmarks.sh` does that, too.

3.1 Garbage Collector

The standalone files come with a precompiled version of the Boehm Garbage Collector (<http://hboehm.info/gc/index.html>). It may not work with a different version of Linux/Ubuntu than the one we used. In this case, you will need to download the garbage collector source and compile it according to your target's needs. Then, replace `libgc.a` and `libcord.a` in `BENCHMARKS/ngship`.