# Language Runtimes

This page gives an overview of upstream projects. If you miss information or find mistakes, please edit.

---

## Dart

Dart is client-optimized programming language.

- Dart homepage

### RISC-V status

- Support for RV32GC and RV64GC

---

## GHC

The Glasgow Haskell Compiler (GHC) is an open-source compiler for the programming language Haskell.

- GHC homepage

### RISC-V status

GHC does not support RISC-V, but there are people working on it. The porting attempts are tracked here:

- https://gitlab.haskell.org/ghc/ghc/-/issues/16783

---

## Go

Go is a programming language designed for simplicity, reliability, and efficiency.

- Go Homepage

### RISC-V status

RISC-V support has been mainlined with Go 1.14.

---

## OpenJDK

OpenJDK is an open-source implementation of the Java Platform (Java SE).

- OpenJDK Homepage

- OpenJDK github repository

### RISC-V status

The OpenJDK RISC-V Port project has been approved by the OpenJDK community [1] and Fei Yang from Huawei Technologies is leading this new project now [2].

They have a project-specific repo for future development here [3].

This port only supports RV64GV for now, shorthand for RV64IMAFDV ISA extensions, and covers the templateInterpreter, C1 and C2, excluding AOT /JVMCI.

All existing GCs are available on riscv64, including ZGC and ShenandoahGC. The VectorAPI and ForeignAPI features are not supported for now. Support for vector

operations is experimental, they need to do more testing for this part.

Aleksey Shipilev is building nightlies here [4]. And they have provided build instructions for reference [5]. This port has passed jtreg tier{1,2,3,4} and jcstress tests

on HiFive Unmatched board. SPECjbb & SPECjvm benchmark tests are also carried out regularly. So it should be good enough to run most Java programs.

A JEP has been submitted in order to bring it upstream in JDK mainline [6], but that process will surely take some time.

[1] https://openjdk.java.net/projects/riscv-port

[2] https://openjdk.java.net/census#riscv-port

[3] https://github.com/openjdk/riscv-port/tree/riscv-port

[4] https://builds.shipilev.net/openjdk-jdk-riscv

[5] http://cr.openjdk.java.net/~fyang/openjdk-riscv-port/BuildRISCVJDK.md

[6] https://openjdk.java.net/jeps/8276797

---

## PHP

PHP is a programming language targeting web development.

- PHP homepage

### RISC-V status

The PHP interpreter works on RISC-V. However, certain features, such as the PCRE JIT, are not ported yet and there are no RISC-V specific optimizations in the codebase at the moment. Thus optimal performance cannot be achieved right now.

---

## Rust

Rust is a programming language designed for performance, reliability and productivity. It relies on LLVM for generating code.

- Rust Homepage
- Rust Book

### RISC-V status

Rust has tier 2 support for RISC-V RV64 (riscv64gc-unknown-linux-gnu) since 1.42.0 (2020-03-12). RV32 is supported, but currenlty does not have a complete standard library.

- https://doc.rust-lang.org/nightly/rustc/platform-support.html

---

# Rust Embedded

Rust Embedded is a project for enabling the Rust Programming Language on "Bare Metal" embedded systems.

- Rust Embedded Working Group
- Embedded Rust Book

## RISC-V status

Since Embedded Rust does not require Rust’s standard library (no_std), more RISC-V platforms are available than for a full Rust runtime.

The following platforms are supported:

- Bare RISC-V RV32
  - RV32I
  - RV32IMAC
  - RV32IMC
- Bare RISC-V RV64
  - RV64IMAFDC
  - RV64IMAC

There is also a quickstart guide to build Rust applications for HiFive1 boards:

- https://github.com/riscv-rust/riscv-rust-quickstart

---

# V8

V8 is a JavaScript and WebAssembly engine with a strong focus on performance.

V8 is used a JavaScript runtime for other runtimes/projects, such as

- Google Chrome Browser
- Chromium Browser
- Node.js

---

# .Net

.Net is a C# runtime. It has two different backend engine.

- CoreCLR Status
- Mono Status

## RISC-V status

The RISCV64GC backend has been upstreamed on Feb 10, 2021 with the joint effort of RIOS lab, Futurewei and PLCT. Now it's maintained by PLCT.

Node.js' support for RISCV64GC has been mainlined with V17.0.0 and now it's maintained by PLCT.

The maintaining page for RISCV V8 related projects can be found in https://github.com/riscv-collab/v8 .