Request for cooperation: Try your applications/libraries with master branch and debug options

If you maintain a Ruby application or library, please consider testing with the Ruby master branch, including the debug build. This should be in addition to testing with all supported stable releases of Ruby.

To make this easier, we are providing Docker images and GitHub actions, as outlined below.

Details

The rapid pace of Ruby development sometimes introduces bugs, such as incorrect behaviour or unexpected incompatibilities. Despite our best efforts and testing, without your feedback, we cannot catch every issue.

Understanding how our changes impact downstream code is important feedback for the Ruby core developers. We want to know how your applications work on master.

If you encounter an error when testing with the master branch (e.g. [BUG] ... in output log), please report it. It will be very helpful.

Testing With master

Testing using the master branch (sometimes referred to as ruby-head) will make your Ruby scripts ready for the next Ruby version. It also helps us catch incompatibilities as we change and evolve Ruby's public interface.

Testing With Debug Build

Testing with the master branch debug build enables many assertions within the Ruby interpreter.

These assertions can detect incorrect usage of the C extensions, and also bugs in the interpreter when running your program.

These assertions have an impact on the performance of the interpreter.

To compile a debug build, refer the later section titled "Building With Debug Mode".

Continuous Integration With master

Building Ruby for your own testing environment can be difficult, so we are providing two convenient ways to use the master branch in your existing testing pipeline:

- Docker Images
- Github Action

Docker Images

The [rubylang docker repository](https://hub.docker.com/r/rubylang/ruby/) provides images for various Ruby versions, including nightly builds of master with and without debug assertions

- Nightly built master: rubylang/ruby:master-nightly-bionic
- Nightly debug built master: rubylang/ruby:master-debug-nightly-bionic

Here is an example Dockerfile:

```docker
FROM rubylang/ruby:master-nightly-bionic
```

Then to build:
$ docker build

Sending build context to Docker daemon 2.048kB
Step 1/1 : FROM rubylang/ruby:master-nightly-bionic
  master-nightly-bionic: Pulling from rubylang/ruby
...
Status: Downloaded newer image for rubylang/ruby:master-nightly-bionic
  --> 059d367a8fbd
Successfully built 059d367a8fbd

GitHub Action

The GitHub Action to setup Ruby provides both ruby-head and ruby-debug builds.

Here is an example workflow to test on all Ruby stable releases, including ruby-head and ruby-debug:

name: Development
on: [push]
jobs:
  test:
    strategy:
      fail-fast: false
    matrix:
      os: [ubuntu]
      ruby: [2.5, 2.6, 2.7, head, debug]
    runs-on: ${{ matrix.os }}-latest
    continue-on-error: ${{ matrix.ruby == 'head' || matrix.ruby == 'debug' }}
    steps:
      - uses: actions/checkout@v2
      - uses: ruby/setup-ruby@v1
        with:
          ruby-version: ${{ matrix.ruby }}
      - run: bundle install
      - run: bundle exec rake

See the documentation for more details on how to use this action.

Building With Debug Mode

To create a debug build of Ruby, execute the following commands:

$ git clone https://github.com/ruby/ruby.git
$ cd ruby
$ autoconf
$ cppflags=--DRUBY_DEBUG=1 ./configure --prefix=$(HOME)/.rubies/ruby-debug
$ make install

If you are using chruby, you can switch to the above build:

$ chruby ruby--debug

You can find more details to build Ruby master in the README.

Acknowledgement

We thank mrkn (Kenta Murata) for the Docker image, Eregon (Benoit Daloze) for the GitHub Action, and ioquatix (Samuel Williams) for reviewing this announcement.

History

#1 - 05/15/2020 02:36 PM - ko1 (Koichi Sasada)
- Description updated
- Subject changed from Request for cooperation: Try your applications/libraries with master and master-debug build to Request for cooperation: Try your applications/libraries with master branch and debug options

03/17/2022
Samuel and Eregon rewrote a description. Thank you!

#2 - 05/15/2020 11:47 PM - ioquatix (Samuel Williams)
- Description updated

Updated from latest edits.

#3 - 05/15/2020 11:51 PM - ioquatix (Samuel Williams)
- Description updated

Make first paragraph read a bit better.