

Ruby and ruby-signature

Goal

- To explain what ruby-signature is and what Ruby3 offers for type checking
- To have a discussion on how it can be integrated with Ruby3
- Align on committers' expectations/motivations to be involved for ruby-signature

What is ruby-signature?

The ruby-signature repository contains two components that will ship with Ruby3; the ruby-signature library and signatures of standard libraries (RBS files). The library consists of Ruby code including a parser of RBS files, type definitions, and utilities on the signatures. The signatures of standard libraries include signatures for builtin classes (Object, Kernel, String, ...) and standard libraries (pathname, set, ...).

ruby-signature

ruby-signature library provides the following functionalities.

- Data definition
 - `Ruby::Signature::AST` - Syntax of RBS files
 - `Ruby::Signature::Types` - Definition of types
- Processing RBS files
 - `Ruby::Signature::Parser` - The parser implemented using Racc
 - `Ruby::Signature::Writer` - Translation from AST to RBS source
- Interpretation of RBS files
 - `Ruby::Signature::Definition` - The class/module definition defined in RBS files
 - `Ruby::Signature::Builder` - Translation from RBS files to Definition
- RBS prototype generators
 - `Ruby::Signature::Prototype::RB` - From Ruby code
 - `Ruby::Signature::Prototype::Runtime` - From Ruby program (runtime)
 - `Ruby::Signature::Prototype::RBI` - From Sorbet RBI
- Testing support
 - `Ruby::Signature::Test`

These functionalities will help the development of:

- Other type checkers,
- Tools to help developers to understand the structure of Ruby programs, and
- RBS file generator for meta-programming.

They read RBS files (parsing) and construct a database to store the structure of Ruby classes/modules (interpretation).

Standard library signatures

Standard library signatures are set of RBS files. We have RBS files for built-in classes and standard libraries.

Currently, we have mechanics to require standard library signatures by adding explicitly options.

```
$ rbs methods -r set Set initialize
```

The `-r set` tells it needs to load the signature for set library.

We expect RBS files to provide:

- List of classes and modules
- List of interfaces, which defines types to describe the constraints of methods and not directly related to Ruby classes
 - Example: `_Each` interface does not have any corresponding Class definition in Ruby programs. It just says "the values have each method". The primary purpose of defining interfaces is to support duck typing.
- Relations between classes and modules (inheritance and mixin)
- List of methods
- List of instance variables

All of the declarations in RBS files can have comments. We are trying to import RDoc comments, and we believe having documentation in RBS will help developers to understand existing libraries.

- Example: <https://github.com/ruby/ruby-signature/blob/master/stdlib/builtin/object.rbs>

How are the type checker developments?

- Sorbet
 - No support for ruby-signature yet
- Steep
 - Supporting ruby-signature
- type-profiler
 - ???

Gem type signatures

- No gems ship with type signatures yet
- Repository to host the signatures by other than gem authors?

- Sorbet has

How can it be integrated with Ruby3?

Expected user experience

Requirements:

1. Ruby has a uniformed/the standard/the official type signature language
 - a. Matz approved
2. Ruby3 users can start using ruby-signature library/type definitions without installing them.
 - a. Matz approved (?)
3. Ruby 2.7 users can install the library/type definition using gem command.
 - a. `$ gem install ruby-signature ruby-signature-stdlib`
 - b. `$ gem install ruby-signature`
4. Ruby3 users can update ruby-signature library/type definitions.
 - a. How often will we update?
5. Help Ruby committers to keep the type definitions up to date.
 - a. If we choose to have the library/type definitions in another repository, Ruby committers won't update them.
6. Ruby3/2.7 users can write type definitions for their programs for some benefits
 - a. How can they have benefits from type definitions?

Discussions

- How it will be distributed
 - Bundled gem?
 - Default gem?
 - Not a gem?
- How it will be developed
 - In ruby/ruby repository?
 - In ruby/ruby-signature repository?
 - Dual upstream?
- What to test with ruby-signature
- How often the library code is updated
 - More frequent than ruby releases?
- How often the type signature updated
 - More frequent than ruby releases?
- What we will do until Ruby3 release
 - Until December 2020
 - Continue development for Ruby 2.7 and test