# RVM 2.0 work announcement

Work on RVM 2.0 has started, check the plan here:

https://docs.google.com/document/d/1xW9GeEpLOWPcddDg\_hOPvK4oeLxJmU3Q5FiCNT7nTAc/edit?usp=sharing

To be able to do the RVM 2.0 work we need to stop working on RVM 1.x -

this means no more feature requests will be accepted.

Development on RVM 1.x will continue to provide bugfixes and

support for new ruby versions as long as RVM 2.0 will not become stable.

RVM is an open source project so any contributions are welcome -

including new features pull requests - as long as they keep compatibility and

do not break anything.

# Common workflows

## Adding support for new version of Ruby

The following files need to be updated to add support for a new version of an already supported Ruby interpreter.

Please follow the patterns used in those files and add entries in the appropriate location.

[\#4366](https://github.com/rvm/rvm/commit/f324ed3946e5d74a80c59d873ac346e70dd50612) is a good example of the required changes.

\* `config/known`

\* update existing entry when minor version released

\* add new entry when major version released

\* `config/known\_strings` - add an entry for the new version released

\* `config/db` - update only for stable releases

\* `config/md5` - add new line with `md5` hash of the archive

\* `config/sha512` - add new line with `sha512` hash of the archive

When listing interpreter source make sure that you link to the smallest archive variant (usually `.bz2`).

When release package does not include `md5` or `sha512` hashes you should download the source package and calculate it yourself.

One of the ways to do that would be to use `openssl` command:

```

openssl dgst -sha512 FILE

openssl dgst -md5 FILE

```

## Running RVM in local development environment

To test a local copy of RVM for development purposes simply enter the folder with the source and install it:

```

./install && rvm reload

```

If you don't want to destroy your working environment, you can select a different destination folder and easily switch between them:

```

./install --path ~/.rvm-dev

rvm switch ~/.rvm-dev # development version

rvm switch ~/.rvm # production version

```

## Subtrees

Subtree is a merge of another repository, it allows us to use external code, track it and simplify its maintenance.

Configuration for subtrees:

```

git remote add bash\_zsh\_support git@github.com:mpapis/bash\_zsh\_support.git

git remote add java\_read\_properties git@github.com:mpapis/java\_read\_properties.git

```

Updating subtrees:

```

git pull --no-commit --no-ff --strategy=subtree bash\_zsh\_support master

git pull --no-commit --no-ff --strategy=subtree java\_read\_properties master

```

It will leave code uncommitted so we can update CHANGELOG.md.

### Adding new subtrees

Follow instructions in official git docs https://www.kernel.org/pub/software/scm/git/docs/howto/using-merge-subtree.html

With the exception of `<2>` needing extra flag `--allow-unrelated-histories`:

```

git merge -s ours --no-commit --allow-unrelated-histories ...

```

## Coding guidelines

https://github.com/rvm/rvm/blob/master/FORMATTING.md