


# Ruby Environment Configurator

The Ruby environment configurator build feature passes Ruby interpreter to all build steps. The build feature adds the selected Ruby interpreter and gems bin directories to the system PATH environment variable and configures other necessary environment variables in case of the RVM interpreter. E.g. in the [Command Line](#) build runner you will be able to directly use such commands as ruby, rake, gem, bundle, etc. Thus if you want to install gems before launching the [Rake](#) build runner, you need to add the [Command Line](#) build step which launches a custom script, e.g.:

```
gem install rake --no-ri --no-rdoc
gem install bundler --no-ri --no-rdoc
```

## Ruby Environment Configurator Settings

Option	Description
Ruby interpreter path	<p>the path to Ruby interpreter. If not specified, the interpreter will be searched in the PATH. In this field you can use values of environment and system variables.</p> <p>For example:</p> <pre>%env.I_AM_DEFINED_IN_BUILDAGENT_CONFIGURATION%</pre>
RVM interpreter	<p>Specify here the RVM interpreter name and optionally a gemset configured on a build agent. Note, that the interpreter name cannot be empty. If gemset isn't specified, the default one will be used.</p> <p> This option can be used if you don't want to use the .rvmrc settings, for instance to run tests on different ruby interpreters instead of those hard-coded in the .rvmrc file.</p>
RVM with .rvmrc file	<p>Specify here the path to a .rvmrc file relative to the checkout directory. If the file is specified, TeamCity will fetch environment variables using the rvm-shell and will pass it to all build steps.</p>
Fail build if Ruby interpreter wasn't found	<p>Check the option to fail a build if the Ruby environment configurator cannot pass the Ruby interpreter to the step execution environment because the interpreter wasn't found on the agent.</p>

See also:

[Administrator's Guide: Configuring Build Steps | Command Line | Rake](#)