


Rake

 TeamCity Rake runner supports Test::Unit, Test-Spec, Shoulda, RSpec, Cucumber test frameworks. It is compatible with Ruby interpreters installed using Ruby Version Manager (MRI Ruby, JRuby, IronRuby, REE, MacRuby, etc.) with rake 0.7.3 gem or higher.

In this section:

- [Prerequisites](#)
- [Important Notes](#)
- [Rake Runner Settings](#)
 - [Rake Parameters](#)
 - [Ruby Interpreter](#)
 - [Launching Parameters](#)
 - [Tests Reporting](#)
- [Known Issues](#)
- [Additional Runner Options](#)
- [Development Links](#)

Prerequisites


Make sure to have Ruby interpreter (MRI Ruby, JRuby, IronRuby, REE, MacRuby, or etc) with rake 0.7.3 gem or higher (mandatory) and all necessary gems for your Ruby (or ROR) projects and testing frameworks installed on at least one build agent.

You can install several Ruby interpreters in different folders. On Linux/MacOS it is easier to configure using [RVM](#). It is possible to install Ruby interpreter and necessary Ruby gems using [Command Line](#) build runner step. If you want to automatically configure agent requirements for this interpreters you need to register its paths in build agent configuration properties and then refer to such property name in [Rake build runner configuration](#).

To install a gem execute:

```
gem install <gem's name>
```

You can refer to the [Ruby Gems Manuals](#) for more information. Also instead of gem command you can install gems using [Bundler gem](#)

 If you use Ruby 1.9 for Shoulda, Test-Spec and Test::Unit frameworks to operate, the 'test-unit' gem must be installed.

Important Notes

- Ruby's pending specs are shown as Ignored Tests in the Overview tab.
- Rake Runner uses its own unit tests runner and loads it using `RUBYLIB` environment variable. You need to ensure your program doesn't clear this environment variable, but you may append your paths to it.
- If you run RSpec with the '--color' option enabled under Windows OS, RSpec will suggest you install the win32console gem. This warning will appear in your build log, but you can ignore it. TeamCity Rake Runner doesn't support coloured output in the build log and doesn't use this feature.
- Rake Runner runs spec examples with a custom formatter. If you use additional console formatter, your build log will contain redundant information.
- `Spec::Rake::SpecTask.spec_opts` of your rakefile is affected by `SPEC_OPTS` command line parameter. Rake Runner always uses `SPEC_OPTS` to setup its custom formatter. Thus you should set up Spec Options in Web UI. The same limitation exist for Cucumber tests options.
- When reporting issues, you can look up version of the Rake Runner in `rakeRunnerPluginAgent.zip/version` (agent plugin), and `rakeRunnerPluginServer.jar/version` (server plugin).
- To include HTML reports into the Build Results, you can add corresponding [report tab](#) for them.

Rake Runner Settings

Rake Parameters

Option	Description
Path to a Rakefile file	Enter Rakefile path if you don't want to use a default one. Specified path should be relative to the build checkout directory .
Rakefile content	Type in the Rakefile content instead of using existing Rakefile. The new Rakefile will be created dynamically from the specified content before running Rake.
Working directory	Optional. Specify if it differs from the build checkout directory .
Rake tasks	Enter task names separated by space character if you don't want to use 'default' task. For example, 'test:functionals' or 'mytask:test mytask:test2'.
Additional Rake command line parameters	Specified parameters will be added to 'rake' command line. The command line will have the following format: <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"><pre>ruby rake <Additional Rake command line parameters> <TeamCity Rake Runner options, e.g TESTOPTS> <tasks></pre></div>

Ruby Interpreter


Option	Description
Use default Ruby	Use Ruby interpreter settings defined in Ruby environment configurator build feature settings or interpreter will be searched in the <code>PATH</code> .
Ruby interpreter path	Path to Ruby interpreter. Path cannot be empty. In this field you can use values of environment and system variables. For example: <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"><pre>%env.I_AM_DEFINED_IN_BUILDAGENT_CONFIGURATION%</pre></div>
RVM interpreter	Specify here the RVM interpreter name and optionally a gemset configured on a build agent. Note, that interpreter name cannot be empty. If gemset isn't specified the default one will be used.

Launching Parameters

Option	Description
Bundle: bundle exec	If your project uses Bundler requirements manager and your Rakefile doesn't load bundler setup script, this option will allow you to launch rake tasks using 'bundle exec' command emulation. If you want to execute 'bundle install' command you need to do it in Command Line step before Rake runner step. Also don't forget to setup Ruby environment configurator build feature to automatically pass Ruby interpreter to command line runner.
Debug	Check the Track invoke/execute stages option to enable showing Invoke stage data in the build log.

Tests Reporting

Option	Description
--------	-------------

Attached reporters	<p>If you want TeamCity to display test results on the dedicated Tests tab of the Build Results page, select here the testing framework you use: Test::Unit, Test-Spec, Shoulda, RSpec or Cucumber.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p> If you're using RSpec or Cucumber, make sure to specify here the user options defined in your build script, otherwise they will be ignored.</p> </div>
--------------------	---

Known Issues

- If your Rake tasks or tests run in parallel in the scope of one build, the build output and tests results will be inaccurate.
- If you are using RVM it is recommended to start TeamCity agent when current rvm sdk isn't set or invoke "rvm system" at first.

Additional Runner Options

These options can be configured using system properties in [Build Parameters](#) section.

Option	Description
system.teamcity.rake.runner.gem.rake.version	Allows to specify which rake gem to use for launching rake build.
system.teamcity.rake.runner.gem.testunit.version	If your application doesn't use latest installed (in Ruby sdk) test-unit gem version please specify it here. Otherwise Test::Unit test reporter may try to load incorrect gem version and affect runtime behavior. If test-unit gem is installed but you application uses Test::Unit bundled in Ruby 1.8.x SDK please set version value to 'built-in'.
system.teamcity.rake.runner.gem.bundler.version	Launches bundler emulation for specified bundler gem version (the gem should be already installed on an agent).
system.teamcity.rake.runner.custom.gemfile	Customizes Gemfile if it isn't located in checkout directory root.
system.teamcity.rake.runner.custom.bundle.path	Sets BUNDLE_PATH if TeamCity doesn't fetch it correctly from <Gemfile containing directory>/bundle/config.

Development Links

Rake support is implemented as an open-source plugin. For development links refer to the [plugin's page](#).