

# RUBYGEMS - RUBY PACKAGE UTILITY

[http://www.tutorialspoint.com/ruby/ruby\\_gems.htm](http://www.tutorialspoint.com/ruby/ruby_gems.htm)

Copyright © tutorialspoint.com

RubyGems is a package utility for Ruby which installs Ruby software packages and keeps them up-to-date.

## Usage Syntax:

```
$ gem command [arguments...] [options...]
```

## Example:

Check to see whether RubyGems is installed:

```
$ gem --version  
0.9.0
```

## RubyGems Commands:

Here is a list of all important commands for RubyGems:

SN	Command with Description
1	<b>build</b> Build a gem from a gemspec
2	<b>cert</b> Adjust RubyGems certificate settings
3	<b>check</b> Check installed gems
4	<b>cleanup</b> Clean up old versions of installed gems in the local repository
5	<b>contents</b> Display the contents of the installed gems
6	<b>dependency</b> Show the dependencies of an installed gem
7	<b>environment</b> Display RubyGems environmental information
8	<b>help</b> Provide help on the 'gem' command
9	<b>install</b> Install a gem into the local repository
10	<b>list</b> Display all gems whose name starts with STRING
11	<b>query</b> Query gem information in local or remote repositories
12	<b>rdoc</b> Generates RDoc for pre-installed gems
13	<b>search</b> Display all gems whose name contains STRING

14	<b>specification</b> Display gem specification <i>inyaml</i>
15	<b>uninstall</b> Uninstall a gem from the local repository
16	<b>unpack</b> Unpack an installed gem to the current directory
17	<b>update</b> Update the named gem <i>orallinstalledgems</i> in the local repository

## RubyGems Common Command Options:

Following is the list of common options:

SN	Command with Description
1	<b>--source URL</b> Use URL as the remote source for gems
2	<b>-p,--[no-]http-proxy [URL]</b> Use HTTP proxy for remote operations
3	<b>-h, --help</b> Get help on this command
4	<b>--config-file FILE</b> Use this config file instead of default
5	<b>--backtrace</b> Show stack backtrace on errors
6	<b>--debug</b> Turn on Ruby debugging

## RubyGems Install Command Options:

This is a list of the options, which use most of the time when you use RubyGems while installing any Ruby package:

SN	Command with Description
1	<b>-v, --version VERSION</b> Specify version of gem to install
2	<b>-l, --local</b> Restrict operations to the LOCAL domain <i>default</i>
3	<b>-r, --remote</b> Restrict operations to the REMOTE domain
4	<b>-b, --both</b> Allow LOCAL and REMOTE operations
5	<b>-i, --install-dir DIR</b> Where to install
6	<b>-d,--[no-]rdoc</b> Generate RDoc documentation for the gem on install

7	<b>-f,--[no-]force</b> Force gem to install, bypassing dependency checks
8	<b>-t,--[no-]test</b> Run unit tests prior to installation
9	<b>-w,--[no-]wrappers</b> Use bin wrappers for executables
10	<b>-P,--trust-policy POLICY</b> Specify gem trust policy
11	<b>--ignore-dependencies</b> Do not install any required dependent gems
12	<b>-y,--include-dependencies</b> Unconditionally install the required dependent gems

## Examples:

This will install 'SOAP4R', either from local directory or remote server including all the dependencies:

```
gem install soap4r --include-dependencies
```

This will install 'rake', only from remote server:

```
gem install rake --remote
```

This will install 'rake' from remote server, and run unit tests, and generate RDocs:

```
gem install --remote rake --test --rdoc --ri
```

## Further Readings:

- The [RubyGems User Guide](#) gives you most everything you need to know about using RubyGems.
- The [RubyGems Command Reference](#) for RubyGems.

Loading [MathJax]/jax/output/HTML-CSS/jax.js