How to Install Ceedling for Windows

Matt Chernosky

www.electronvector.com

Install the required tools

Ceedling requires both Ruby and GCC to run — so we'll need to install those first.



We need to install a native GCC for our host PC, not any kind of cross-compiler for a target processor. Our goal here is to set up to run unit tests on your host PC.

Ceedling will be run from the command line, and so you'll also need to make sure that your path is set up correctly to point to both Ruby and GCC.

Install Ruby

Ruby can be installed for Windows from rubyinstaller.org. You may want to check out the information on the downloads page about which version of Ruby to install. I had plenty of success with the 2.3.x versions of Ruby.

Download the correct installer for your platform—either the 32-bit or the 64-bit (x64) version—and run the installer.

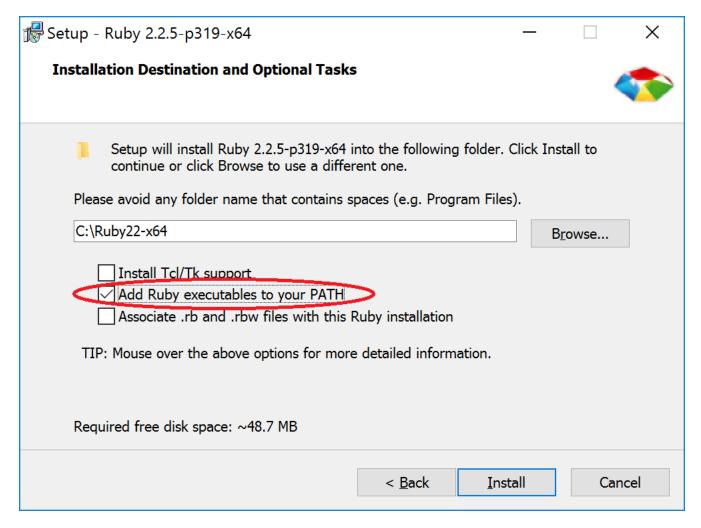
During installation, I recommend using the default installation folder and selecting to have Ruby added to the path. This will make it easier to install and use Ceedling from the command line.



Don't install Ruby into a location with spaces in the path like **C:\Program Files**.



Be sure to select the option for Add Ruby executables to your PATH.



When the installer has finished, you can confirm that the installation was successful by opening up a command prompt and running ruby --version:

```
> ruby --version
ruby 2.3.1p112 (2016-04-26 revision 54768) [x64-mingw32]
```

This shows that we're running with our expected Ruby version.



If you already have a command prompt open — you'll need to make sure you open a *new* command prompt *after* you've finished installing Ruby. If not then your PATH variable won't be set correctly yet.

Install GCC

GCC is the native C compiler used for Ceedling. There are two primary ways to install GCC on Windows: via *MinGW* or *Cygwin*. I recommend *MinGW* for use with Ceedling.

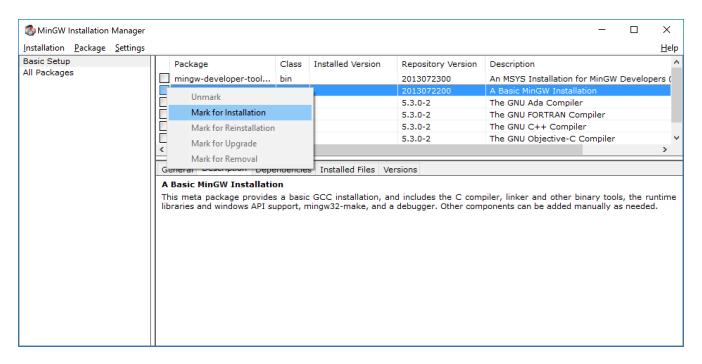
You can get *MinGW* from www.mingw.org via the MinGW installer.

Download and run the installer. The default options should work fine.

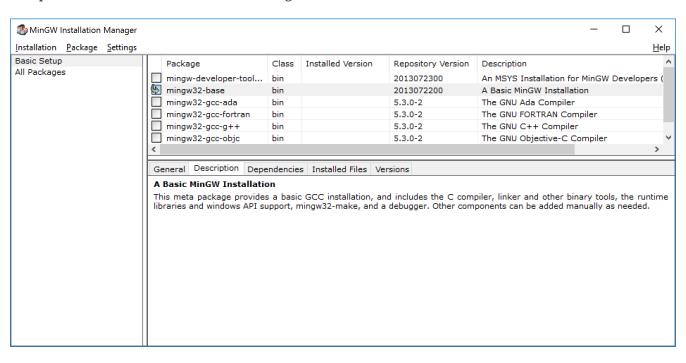


Don't install MinGW into a location with spaces in the path like **C:\Program Files**.

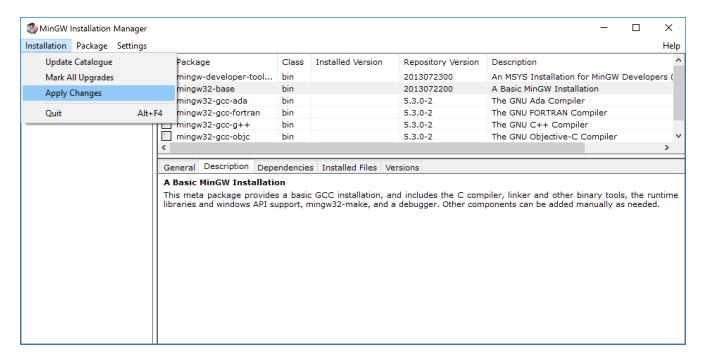
Eventually you'll come to the package selection screen. Click on the box next to **mingw32-base** and select **Mark for Installation**:



The provides a basic installation including GCC:



Click on **Installation** → **Apply Changes**:



Then click the **Apply** button and everything will be installed. When it's done, you can close the installer.

If you need any more help installing MingGW, see the Getting Started guide at mingw.org.

Install Ceedling

You can install Ceedling though the Ruby package manager gem. The gem command should be available once Ruby is installed and available in your path. Just run:

```
> gem install ceedling
```

That's it.

You can check your installed version of Ceedling with:

```
> gem list ceedling
ceedling (0.28.1)
```

And confirm that Ceedling is installed by running it:

```
> ceedling
Welcome to Ceedling!
Commands:
   ceedling example PROJ_NAME [DEST] # new specified example project (in DEST...
   ceedling examples # list available example projects
   ceedling help [COMMAND] # Describe available commands or one spe...
   ceedling new PROJECT_NAME # create a new ceedling project
```

Ceedling Installation Error: "certificate verify failed"

I've recently had an issue trying to use the gem command to install Ceedling. It's a problem with the RubyGems server (where Ceedling and other gems are hosted). If you get an error like this:

ERROR: Could not find a valid gem 'ceedling' (>= 0), here is why:
 Unable to download data from https://rubygems.org/ - SSL_connect returned=1
errno=0 state=SSLv3 read server certificate B: certificate verify failed
(https://api.rubygems.org/specs.4.8.gz)

The quick fix is to install a new certificate into Ruby. Download this certificate file:

https://raw.githubusercontent.com/rubygems/rubygems/master/lib/rubygems/ssl_certs/index.rubygems.org/GlobalSignRootCA.pem

and place it into **C:\Ruby23\lib\ruby\2.3.0\rubygems\ssl_certs**. Note that your path may be different if you installed to another folder or installed a different version.