

Setting Up the development environment for Ruby on Rails -- for Windows 7 without using Cygwin --

Lecture Material

for Object Oriented WEB Programming using Ruby, 2012,
with the versions on May 15, 2012

When we set up Ruby on Rails on windows 7, the procedures differ from the case with Cygwin and without Cygwin. So, the first course material had been written for those Computers with Cygwin environment of the juniors' model of Hosei University. However, (probably,) when the development environment for Android OS applications had been installed, Cygwin did not support 'rvm' installation from gems, and such problems occurred. We may be able to avoid those problems by changing the configuration of tools, but to avoid spending time in vain (from trials and errors,) here the course lecture material to set up Ruby On Rails without Cygwin Environment.

Step 1: Install msysgit

Connect to the repository of "msysgit."

<http://code.google.com/p/msysgit/downloads/list>

Here we choose Git-1.7.10-preview20120409.exe. Download and execute.

| Filename | Summary + Labels | Uploaded | ReleaseDate | Size | Download |
|--|---|----------|-------------|---------|----------|
| Git-1.7.10-preview20120409.exe | Full installer for official Git for Windows 1.7.10 Featured Beta | Apr 9 | | 14.6 MB | |
| PortableGit-1.7.10-preview20120409.7z | Portable application for official Git for Windows 1.7.10 Beta Featured | Apr 9 | | 19.6 MB | |
| msysGit-netinstall-1.7.10-preview20120409.exe | Net installer if you want to hack on Git Beta Featured | Apr 9 | | 2.7 MB | |
| msysGit-fullinstall-1.7.10-preview20120409.exe | Full installer (self-contained) if you want to hack on Git Beta Featured | Apr 9 | | 39.8 MB | |
| Git-1.7.9-preview20120201.exe | Full installer for official Git for Windows 1.7.9 | Feb 1 | Feb 1 | 14.4 MB | |

This is the installation tool of Git for Windows.

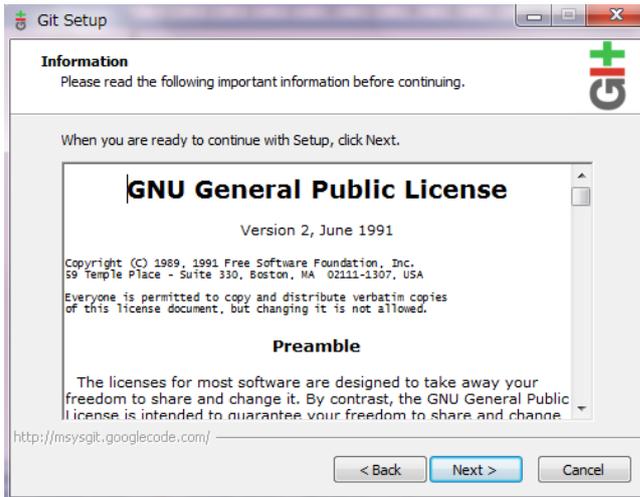


There would be several confirmation if you could execute the programs downloaded from Internet Environment, please answer [Yes] to permit the execution. When the Installation is invoked, the following screen would appear.

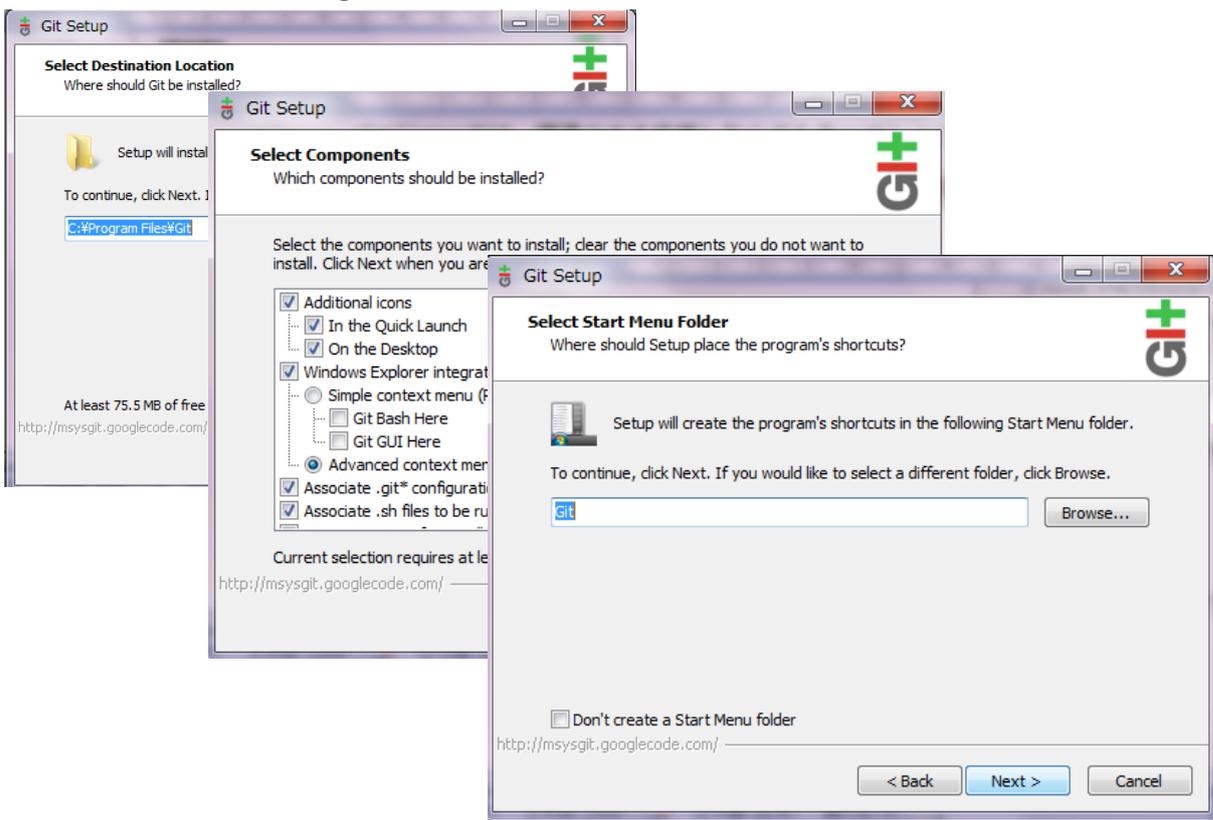


Simply click 'Next.'

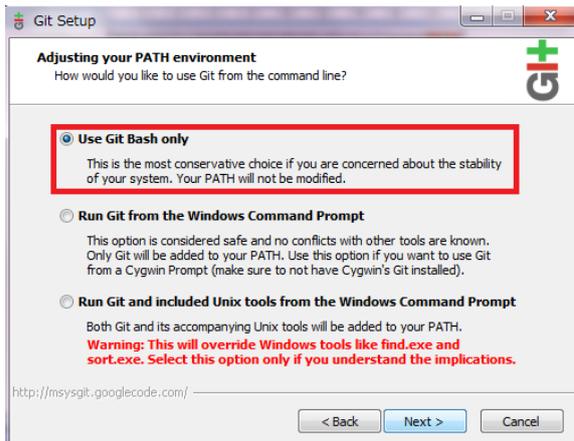
Click 'Next' for the GNU License agreement screen, after reading it through.



We invoke git via GUI (by clicking desktop icon), and we may not start it from command prompt. Therefore, we do not change the installation path from the default settings.



Not change from default settings. Simply click 'Next' for above pop-ups. Register to Start Menu with the name of 'Git'. I have not tried the multi-git environment, so if you need to distinguish those gits clearly, use the name of 'msysgit' for Start Menu registration.



For PATH setting, give check mark to 'Use Git Bash Only.'

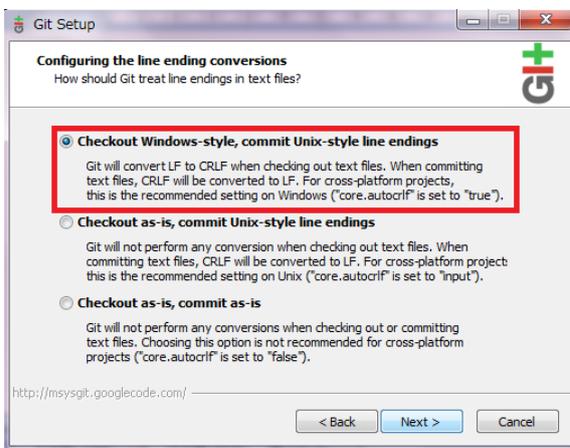
Especially here, I am scared of competition between msysgit and Cygwin, (especially for Android Application Development Environment,) so, when it had come to the situation we had to change the PATH for msysgit, we do it manually, set up tools manually, and we do not change PATH automatically.

* In case you freely install Msysgit for the Windows 7 without Cygwin environment, it would be better using the option appeared the below most, (i.e. modify PATH, and use UNIX tools also.) For this case, our policy is to install rails via gems without using 'DevKit,' and we use DevKit only when the following errors appeared in the command line messages.

ERROR: Error installing rails:

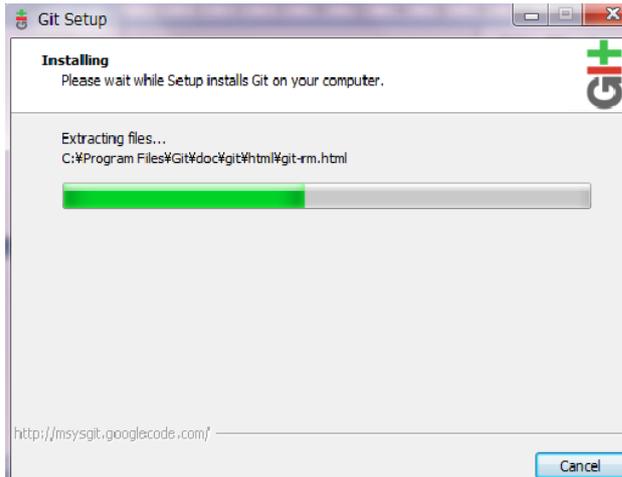
The 'json' native gem requires installed build tools.

This course material is written on condition that there is Cygwin environment, but we do not use Cygwin for Ruby on Rails.

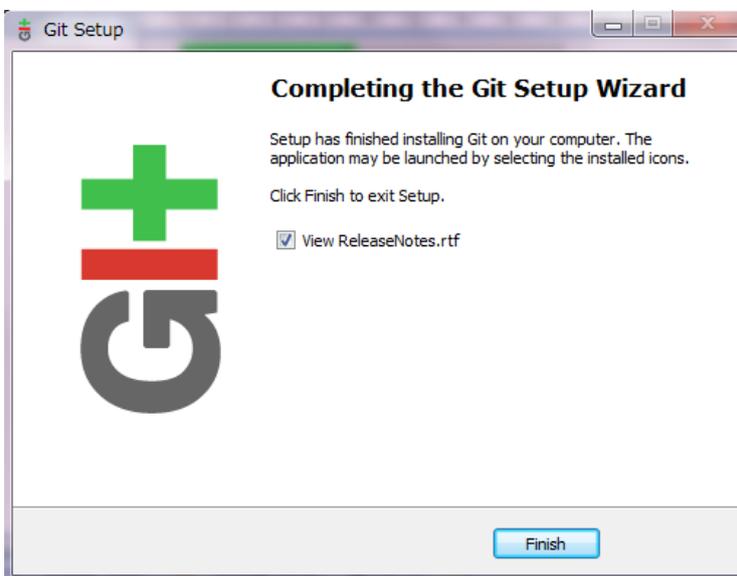


For next option, we choose default option for using 'CRLF,' shown in the top most.

This is for using this Windows 7 PC as a server in this course, so that we could read and edit the text files easily on this Windows 7 environment.



Wait until the installation is finished.



If you see this screen, the installation is done.

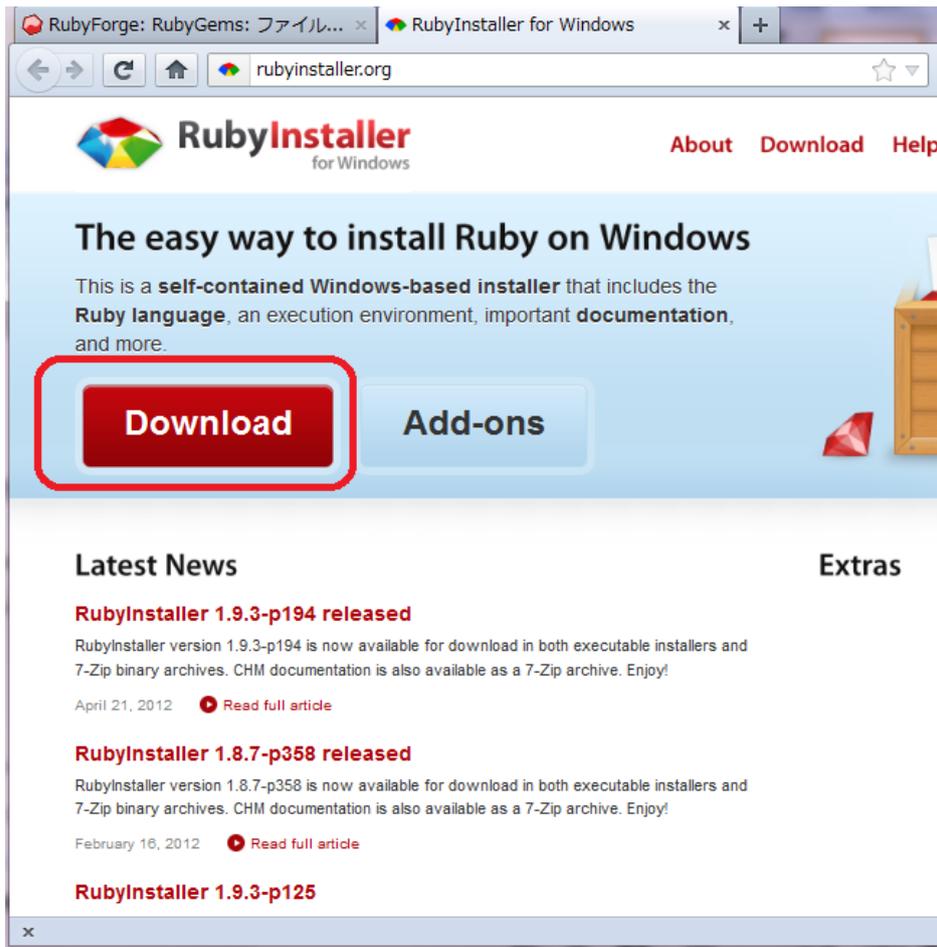
This course memo is written on condition the Cygwin is installed, so if you need to install Ruby on Rails with Msysgit, please decide the tools carefully, depending

which kinds of tools are installed on that computer.

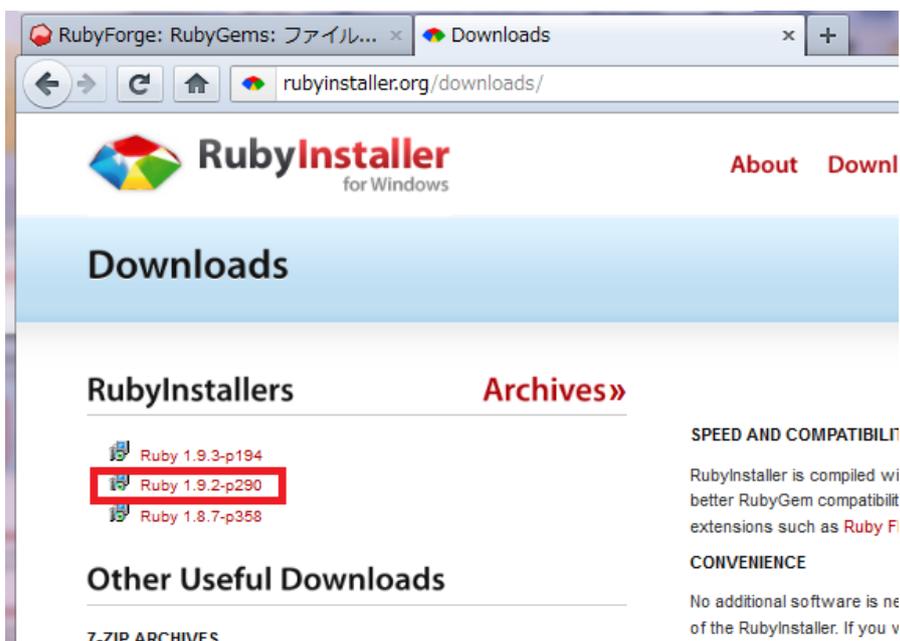
Step 2: Install Ruby

Ruby is already installed in Cygwin Environment, but we want to disconnect Cygwin and this Msysgit environment, we install ruby 1.9.2 independently from Cygwin for Windows 7. First, connect to the following site:

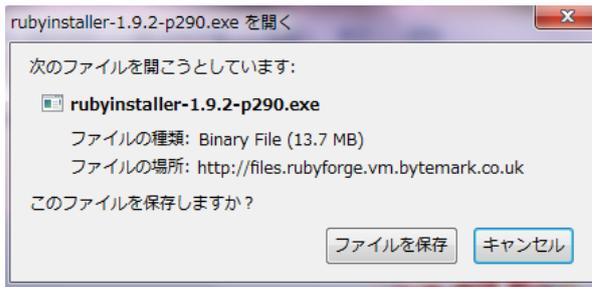
<http://rubyinstaller.org/>



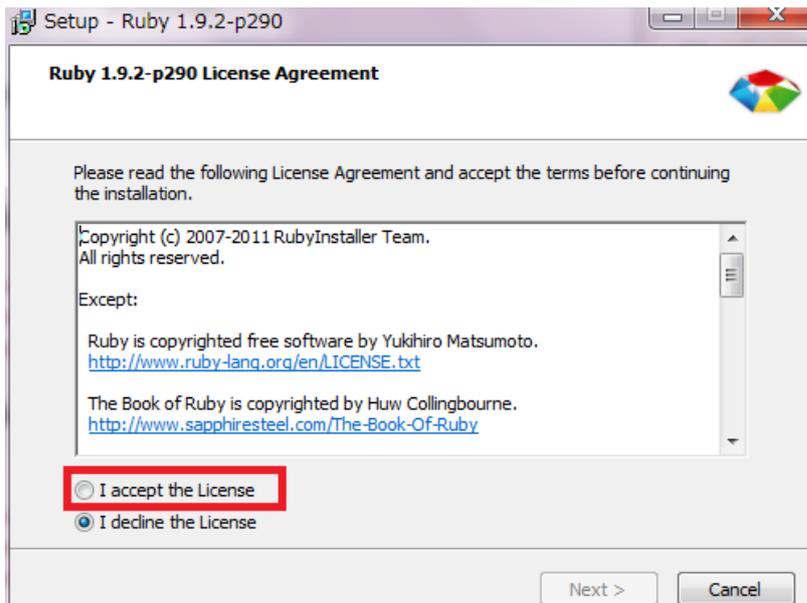
Click the most conspicuous Download button.



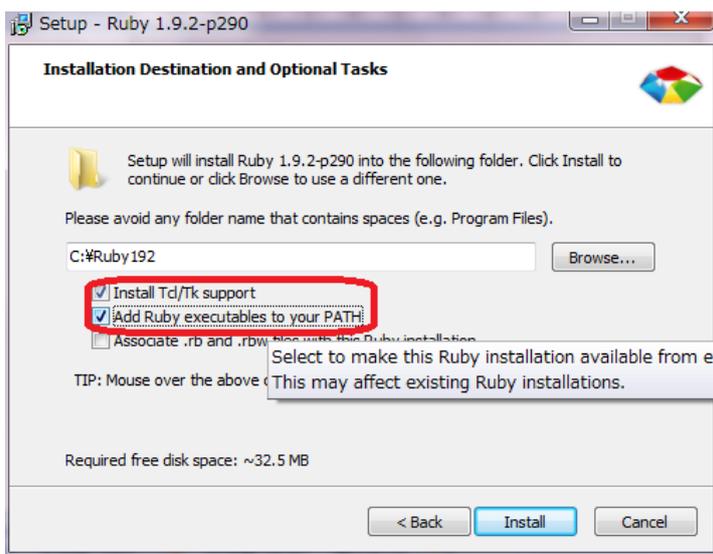
The latest version 1.9.3, but other materials are written on the basis of using 1.9.2, so choose Version 1.9.2.



Download and execute.



Accept the license, and go Next.

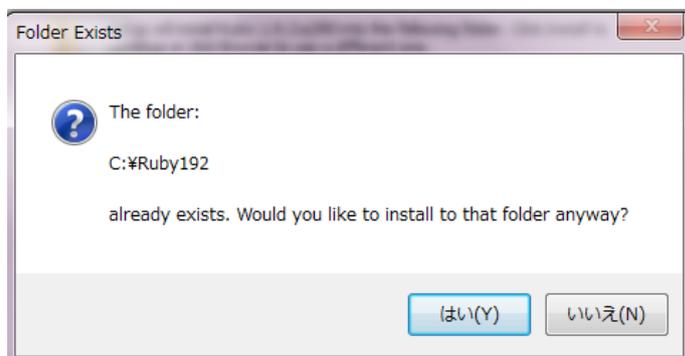


We click Tcl/tk and add ruby PATH to the system environment variable.
The reason is that we do want to tell Cygwin that there is msysgit, but we

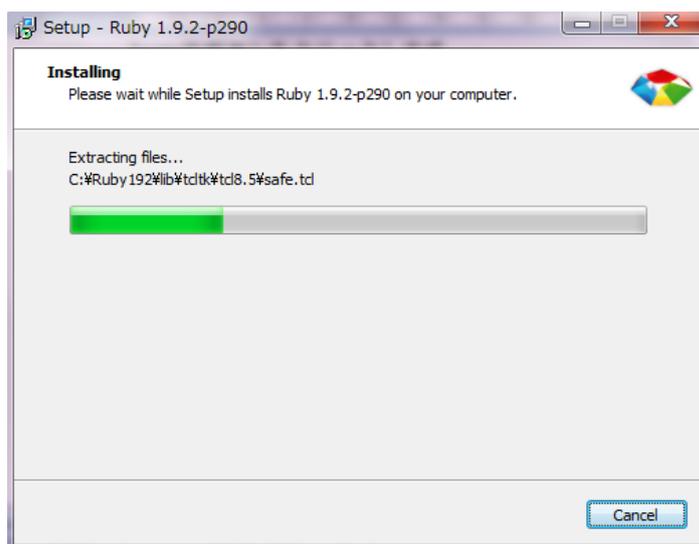
want to tell the presence of ruby1.9.2 to msysgit, and also we (may) have some chance to use tcl/tk during the build and installation.

- There is a warning message on the screen, this PATH setting may affect other development environment. In case this may badly affect Android Emulator environment, remove the system PATH of ruby binary manually, and add C:\Ruby192\bin path manually to git bash environment.
- If you failed to install with this memo, (or badly affect other development projects,) please consult the lecturer or TA.

Click install button.



The installer ask you to create the ruby192 folder, answer “YES.”



Now the installation starts.



If you see this popup, the installation of Ruby 1.9.2 finished.

Step 3: (Skip) installation of MinGW

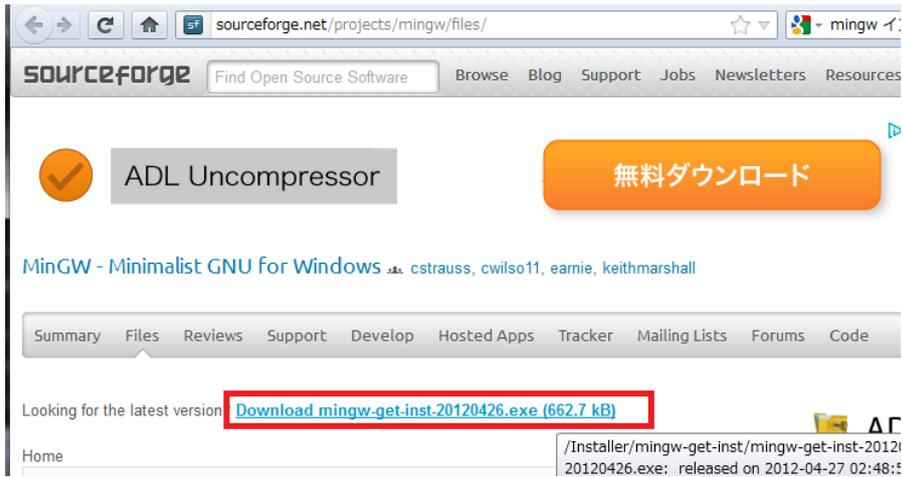
I leave the following memo with my first trial of installation. Step 4 is the installation of DevKit, and there is MinGW included in DevKit, so it is not needed to install MinGW separately. So we skip this procedure.

Just in case you failed to install DevKit, please follow this 'Step 3', to install MinGW.

Currently, I am not certain if we need MinGW installation for our purpose. The reason why the procedure is left in this material is I could have passed through the compile errors on the system environment both Android development and Cygwin for ruby. So, this may be one of the hints to solve the installation problem. Connect to the following site.

<http://sourceforge.net/projects/mingw/files/>

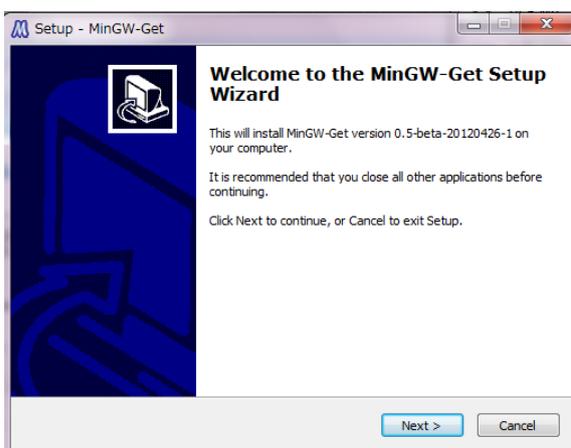
If you have installed MinGW for Android Application development environment already, this Step 3 is not needed.



Simply download the latest version.

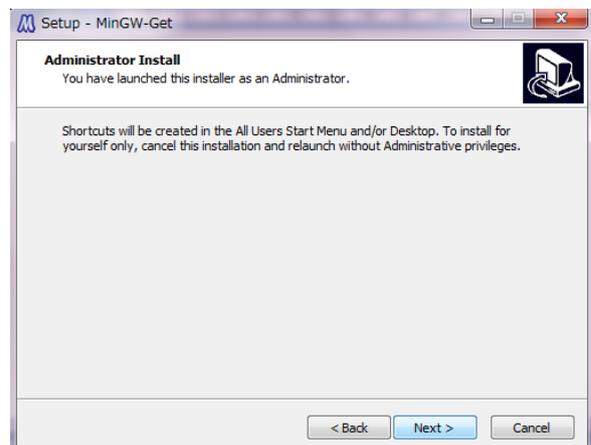


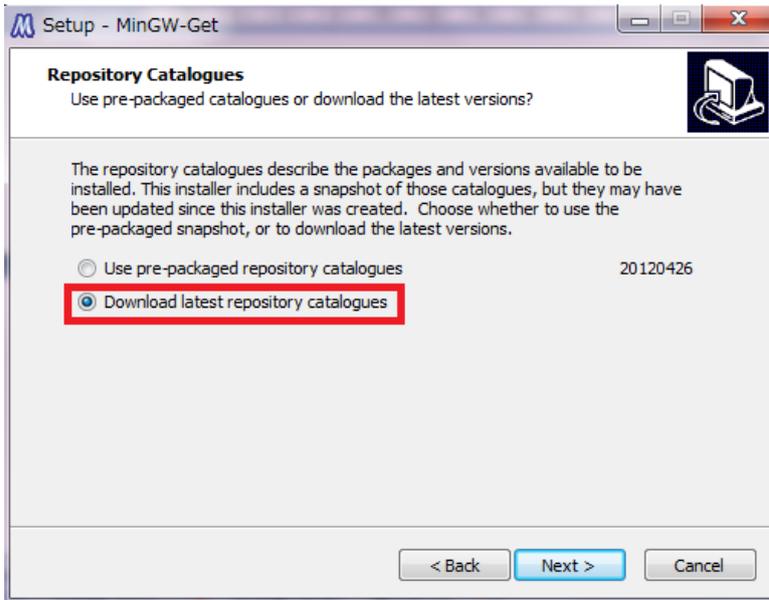
After waiting for a moment, you would see the file save popup. Download and execute.



Click 'Next' here.

Click 'Next' here, and

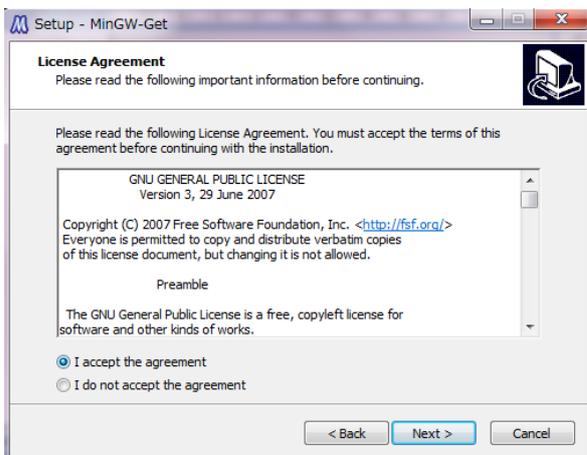




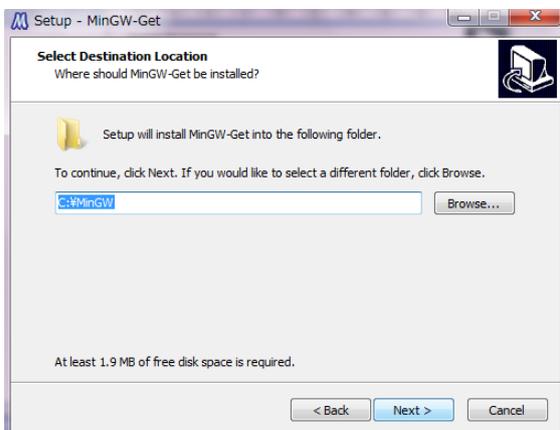
Check to download the latest catalog from the repository.

But if you have any anxiety about the consistency with other tools, (for my case, I had installed MinGW, then removed, several times, so you may not

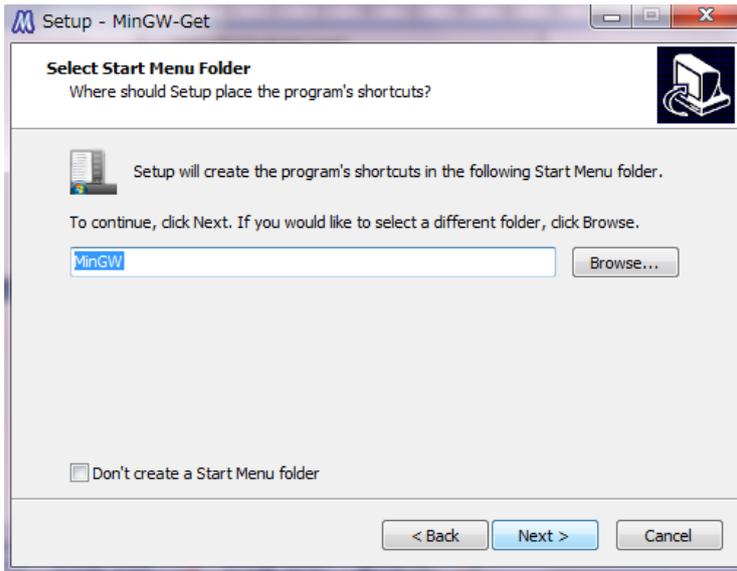
face with this selection, though,) you may choose to use the pre-packaged repository catalogues, or even you may need not to install MinGW.



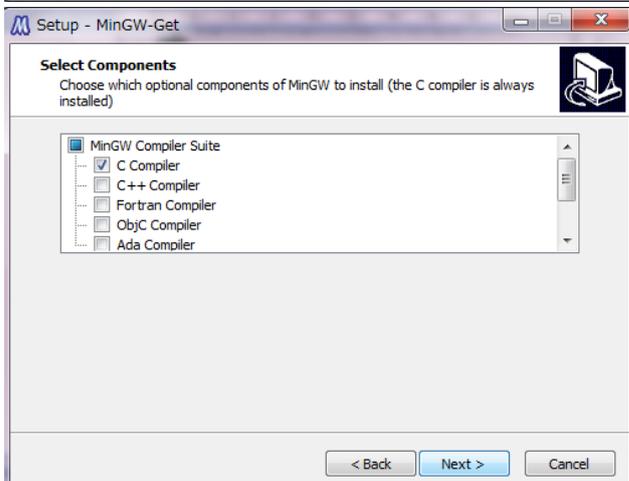
Please agree with the license condition.



Install it to C:\MinGW.

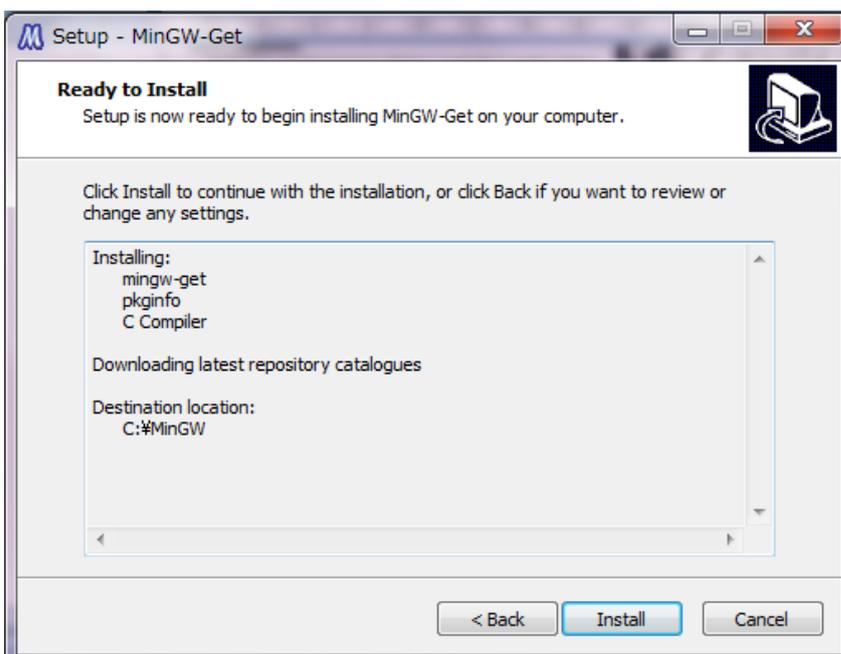


Click 'next'.



Choose only C compiler.

If you want to enjoy Ada and such classical programming languages, there would be no harm by doing so.



Click 'Install'

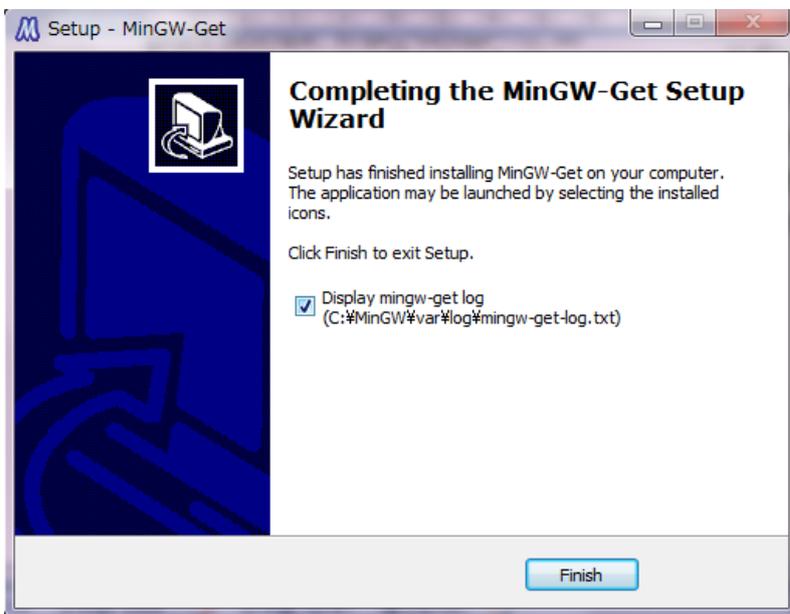
```
C:\Windows\system32\cmd.exe
Update catalogue: package-list.xml
Update catalogue: mingw32-package-list.xml
Update catalogue: mingw32-autoconf.xml
Update catalogue: mingw32-automake.xml
Update catalogue: mingw32-basic-bsdtar.xml
Update catalogue: mingw32-binutils.xml
Update catalogue: mingw32-bzip2.xml
```

Download immediately, and start the repository update screen.

Just wait.

It depends on the communication speed, but

you have to wait only several minutes to 20-30 minutes at most...



If you saw this popup message, it finished.

Step 4: Install DevKit

The reason why we need to install DevKit is described in the following site: <http://supportdoc.net/support-ror3/devkit.html>

We could know the necessity of installing DevKit with the following error message;

Please update your PATH to include build tools or download the DevKit from 'http://rubyinstaller.org/downloads' and follow the instructions at 'http://github.com/oneclick/rubyinstaller/wiki/Development-Kit'

As the result, we need to install DevKit. Download the file;

DevKit-tdm-32-4.5.2-20111229-1559-sfx.exe

from the site;

<http://rubyinstaller.org/downloads>

* Please note, that we are going to use ruby 1.9.2, so the DevKit must be the version 4.5.2, from the following message.

NOTE: for the legacy RubyInstaller v1.8.6 use the DevKit-3.4.5 available at our archive downloads page. For RubyInstaller versions 1.8.7, 1.9.2, and 1.9.3 use the DevKit 4.5.2 from our main downloads page.

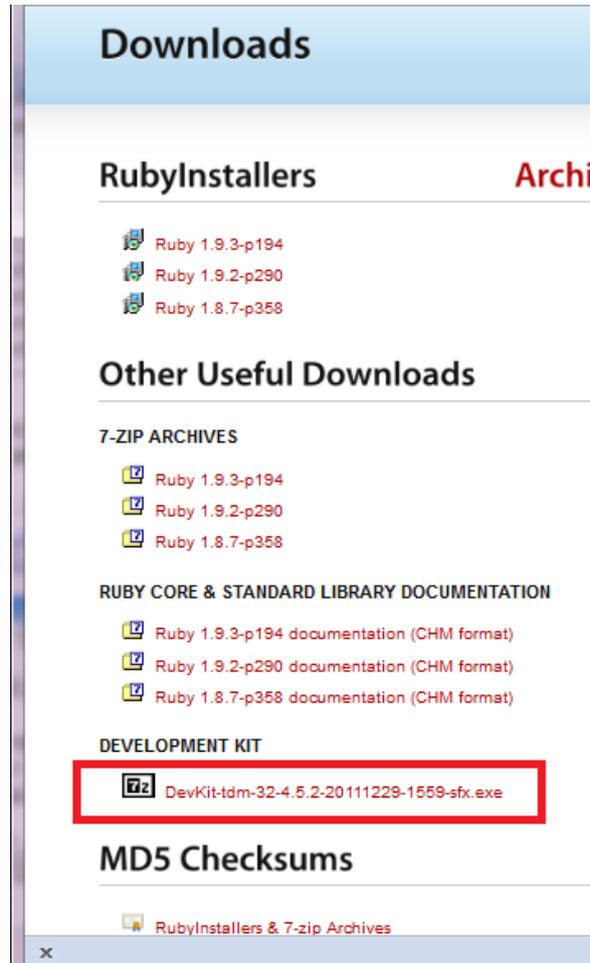
If you had already installed other C compiler environment from MinGW or other tools, the message above may not shown. There could be the option that you install DevKit only when you faced with the above error message, and skip this step 4.

When you install DevKit, you would find the DevKit Install in the same screen where we obtained the Ruby installer.

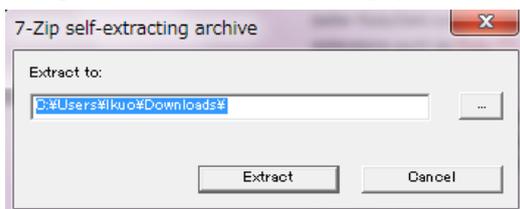
Download the file,

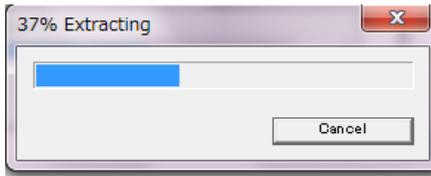
DevKit-tdm-32-4.5.2-20111229-1559-sfx.exe

and run.



It is a self decompression file, so if you start running, decompression starts. But it develop files WITHOUT CREATING parent folder, so please specify the parent (new) folder by yourself. Here we give the name of 'DevKit.'





When decompression ended, open the parent folder. Change Directory (cd) to the given folder. You can use bash of msysgit.

```
Ikuo@DELL-E5500-KOBA ~
$ cd downloads

Ikuo@DELL-E5500-KOBA ~/downloads
$ cd devkit

Ikuo@DELL-E5500-KOBA ~/downloads/devkit
$ ruby dk.rb init
[INFO] found RubyInstaller v1.9.2 at C:/Ruby192

Initialization complete! Please review and modify the auto-generated
'config.yml' file to ensure it contains the root directories to all
of the installed Rubies you want enhanced by the DevKit.

Ikuo@DELL-E5500-KOBA ~/downloads/devkit
$ ruby dk.rb install
[INFO] Updating convenience notice gem override for 'C:/Ruby192'
[INFO] Installing 'C:/Ruby192/lib/ruby/site_ruby/devkit.rb'

Ikuo@DELL-E5500-KOBA ~/downloads/devkit
$
```

In the 'Devkit' developed folder, run the following command.

```
ruby dk.rb init
```

If you receive the message;

```
Initialization complete!
```

then, run the following command.

```
ruby dk.rb install
```

It will finish shortly.

Step 5: Install Rubygems

Next step is to install rubygems.

Here we follow the procedure to download the source codes of rubygems, and compile(build) them, and dispatch the tools to ruby environment.

Start git bash. Usually, when we do things in the UNIX environment, we

```
$ cd /usr
Ikuo@DELL-E5500-KOBA /usr
$ mkdir local

Ikuo@DELL-E5500-KOBA /usr
$ cd local

Ikuo@DELL-E5500-KOBA /usr/local
$ pwd
/usr/local

Ikuo@DELL-E5500-KOBA /usr/local
```

create the working directory in /usr/local, but this msysgit-bash environment, when we create a directory 'local' under /usr directory, the entity of the directory is created in (assume that the user is 'admin',)

/Users/admin/AppData/Local/VirtualStore/Program Files/Git/Local

It is rather deep from the root directory of C:¥ drive of Windows.

Then, we create
rubygems_src directory in
/usr/local directory.

```
Ikuo@DELL-E5500-KOBA /usr/local
$ mkdir rubygems_src

Ikuo@DELL-E5500-KOBA /usr/local
$ cd rubygems_src

Ikuo@DELL-E5500-KOBA /usr/local/rubygems_src
$
```

Type,

cd /usr/local

to move the directory to /usr/local, then type

mkdir rubygems_src

to create the directory which will contain the source codes of ruby gems.

Then type,

cd rubygems_src

to move to the source code directory, and run the command

wget <http://rubyforge.org/frs/download.php/76073/rubygems-1.8.24.tgz>

If your system complains against you, that it does not have wget, then use curl instead.

curl -O <http://rubyforge.org/frs/download.php/76073/rubygems-1.8.24.tgz>

Ehter the above two commands are the ordinary method, but not knowing the reason, both of the above two commands does not work in some PC environment. (I guesst the network constraint may block the connection to the server.)

At last, we may download the file directly from the site, (simply connect to)

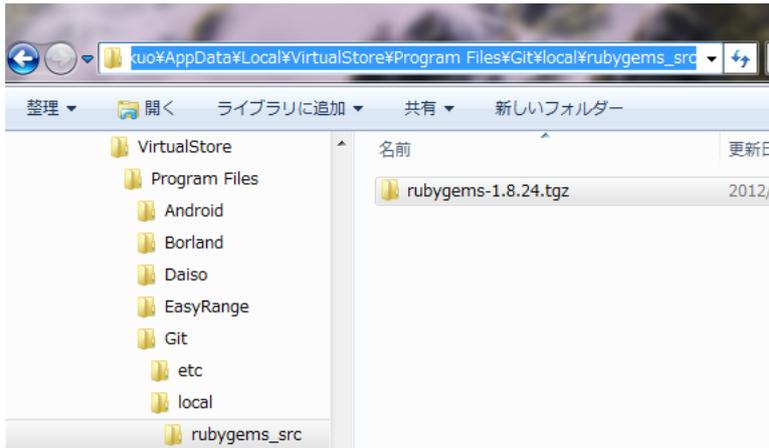
<http://rubyforge.org/frs/download.php/76073/rubygems-1.8.24.tgz>

and download the file, then, move the file into

/Users/admin/AppData/Local/VirtualStore/Program Files/Git/Local/

We follow do the above detour on the Windows 7 environment. It is far more smoother than to do them in Linux/Unix Shell environment.

Use the explorer of Windows.



With the network environment in Koganei Campus of Hosei University, port is not open for git connection, so git commands will not work.

In this case too, we may download the file directly from github, and copy the file directly to the local directory to bypass the git port.

Incidentally, with the network environment in Chiba Campus of Teikyo Heisei University, proxy server blocks the gem connection, so we may need to set the system environment variable of the followings;

```
set http_proxy=http://(URL of Proxy server):(port number of the proxy)
```

```
set http_proxy_user=proxy registered user name
```

```
set http_proxy_pass=proxy registered user password
```

to use gems. For example, for my case,

```
set http_proxy=http://proxy1.thu.ac.jp:80
```

```
set http_proxy_user=i-kobaya
```

```
set http_proxy_pass=myspassword
```

to run gems for ruby. If you face with 'Proxy' problems, please use this hint.

So much for digressions,

```
admin@ADMIN-501MOKPBO /usr/local/rubygems_src
$ wget http://rubyforge.org/frs/download.php/75952/rubygems-1.8.21.tgz
```

or some other detours might be done, get back to the point to start copying files.

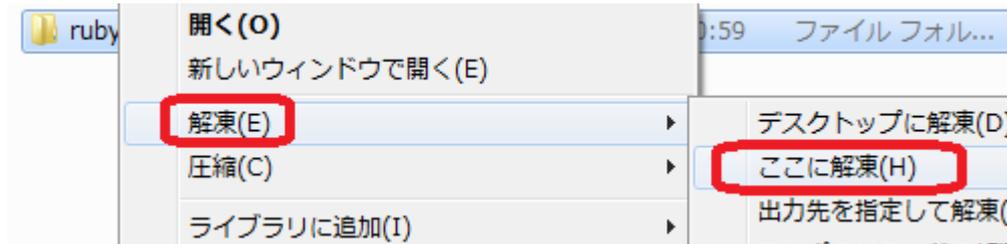
If 'wget' or 'curl' download succeeded, the command to decompress the file is,

```
tar xvzf rubygems-1.8.21.tgz
```

```
admin@ADMIN-501MOKPBO /usr/local/rubygems_src
$ ls
rubygems-1.8.21.tgz

admin@ADMIN-501MOKPBO /usr/local/rubygems_src
$ tar xvzf rubygems-1.8.21.tgz
```

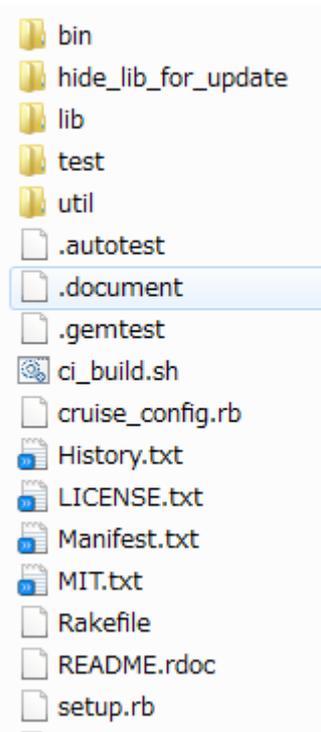
If you download and copied the file in the Windows Environment, you can simply right click the file to decompress the file in the given directory.



(Sorry for the Japanese screen, but please guess what are they!)

When you start tar command, the decompressed folders and files will be shown in the command window.

```
$ tar xvzf rubygems-1.8.24.tgz
rubygems-1.8.24/
rubygems-1.8.24/.autotest
rubygems-1.8.24/.document
rubygems-1.8.24/.gemtest
rubygems-1.8.24/bin/
(omitted)
```



If you see them in the Windows folder, just like it is show in the right.

If you finish developing the source codes, then start the set up.

Once again, just make sure the check the curl existence.

```
curl --version
```

You should do the following procedure where your computer is connected to the network. If curl did not reply, then check the Step 4 DevKit installation or Step 3 MinGW setup.

```
Ikuo@DELL-E5500-KOBA /usr/local/rubygems_src
$ curl --version
curl 7.21.1 (i686-pc-mingw32) libcurl/7.21.1 OpenSSL/0.9.8r zlib/1.2.3
Protocols: dict file ftp ftps http https imap imaps ldap ldaps pop3 pop3s rtsp s
mtp smtps telnet tftp
Features: Largefile NTLM SSL SSPI libz

Ikuo@DELL-E5500-KOBA /usr/local/rubygems_src
$ cd rubygems-1.8.24

Ikuo@DELL-E5500-KOBA /usr/local/rubygems_src/rubygems-1.8.24
$ ls
History.txt  Manifest.txt  UPGRADING.rdoc  cruise_config.rb  setup.rb
LICENSE.txt  README.rdoc  bin             hide_lib_for_update  test
MIT.txt      Rakefile     ci_build.sh    lib                util

Ikuo@DELL-E5500-KOBA /usr/local/rubygems_src/rubygems-1.8.24
$ ruby --version
ruby 1.9.2p290 (2011-07-09) [i386-mingw32]

Ikuo@DELL-E5500-KOBA /usr/local/rubygems_src/rubygems-1.8.24
$ ruby setup.rb
```

Change directory to rubygems-1.8.24

```
cd rubygems-1.8.24
```

then run

```
ruby setup.rb
```

With the above attached screen shots, the command 'ls' is to show the list of the contents of directory, and 'ruby --version' is just to make sure the ruby is installed properly.

When you finished the setup shortly, the following messages will be shown.

```
Ikuo@DELL-E5500-KOBA /usr/local/rubygems_src/rubygems-1.8.24
$ ruby setup.rb
RubyGems 1.8.24 installed

== 1.8.24 / 2012-04-27

* 1 bug fix:

  * Install the .pem files properly. Fixes #320
  * Remove OpenSSL dependency from the http code path

-----

RubyGems installed the following executables:
  c:/Ruby192/bin/gem

Ikuo@DELL-E5500-KOBA /usr/local/rubygems_src/rubygems-1.8.24
```

Confirm that the version number will given with the command;

```
gem -v
```

Step 6: Install rake

Rake is the build management tool for ruby environment.

We install rake with gem.

Type the following command;

```
gem install rake
```

```
admin@ADMIN-501MOKPBO /usr/local/rubygems_src/rubygems-1.8.21
$ gem install rake
Fetching: rake-0.9.2.2.gem (100%)
Successfully installed rake-0.9.2.2
1 gem installed
Installing ri documentation for rake-0.9.2.2...
Installing RDoc documentation for rake-0.9.2.2...
admin@ADMIN-501MOKPBO /usr/local/rubygems_src/rubygems-1.8.21
^
```

If the installation finished shortly, type the following command to check the version,

```
rake --version
```

Step 7: (Skip) install rvm

We usually use rvm to control the ruby version, using git. However, with the composition shown in this memo, it seems that rvm does not work well. So please skip to the Step 8 pik install.

If there is any necessity to use rvm, the explanation is intentionally left described in this memo. Usually, please skip the following;

Move directory to /usr/local with the following command,

```
cd /usr/local
```

Installation command of rvm is

```
git clone --depth 1 git://github.com/wayneeseguin/rvm.git
```

```
admin@ADMIN-501MOKPBO /usr/local
$ git clone --depth 1 git://github.com/wayneeseguin/rvm.git
Cloning into 'rvm'...
remote: Counting objects: 5901, done.
remote: Compressing objects: 100% (3031/3031), done.
Receiving objects: remote: Total 5901 (delta 3926), reused 3953 (delta 2202)98
(5783/5901)
Receiving objects: 100% (5901/5901), 2.02 MiB | 272 KiB/s, done.
Resolving deltas: 100% (3926/3926), done.
admin@ADMIN-501MOKPBO /usr/local
^
```

When the download is completed, then start installation. Move to rvm directory which is generated in /usr/local, by typing the following command,

```
cd rvm
```

Here type the following command,

./install

```
admin@ADMIN-501MOKPBO /usr/local
$ cd rvm

admin@ADMIN-501MOKPBO /usr/local/rvm
$ ./install

Installing RVM to /home/admin/.rvm/
  Adding rvm PATH line to /home/admin/.bashrc /home/admin/.zshrc.
  Adding rvm loading line to /home/admin/.bash_login /home/admin/.zlogin.

# RVM: Shell scripts enabling management of multiple ruby environments.
# RTFM: https://rvm.beginrescueend.com/
# HELP: http://webchat.freenode.net/?channels=rvm (#rvm on irc.freenode.net)
# Cheatsheet: http://cheat.errtheblog.com/s/rvm/
# Screencast: http://screencasts.org/episodes/how-to-use-rvm

# In case of any issues read output of 'rvm requirements' and/or 'rvm notes'
```

When Android Application Development Environment exists on the PC, it may happen to see the following error messages, and I guess competition of the path to C compilers. (I have not investigated well. Sorry.)

```
Ikuo@DELL-E5500-KOBA /usr/local/rvm (master)
$ ./install
scripts/install: line 7: conditional binary operator expected
scripts/install: line 7: syntax error near `=~'
scripts/install: line 7: `if [[ "$@" =~ --trace ]] || (( ${rvm_trace_flag:-0} >
0 ))'

Ikuo@DELL-E5500-KOBA /usr/local/rvm (master)
```

If you had faced with the above message, please discard to setup rvm, and move to the next Step 8.

When you success, the following message will be shown.

```
* To start using RVM you need to run `source /home/admin/.rvm/scripts/rvm`
  in all your open shell windows, in rare cases you need to reopen all shell w
indows.

* Optionally you can run `rvm tools rvm-env ruby bash` which will generate
  shebang wrappers for easier selecting ruby in scripts.

# admin,
#
# Thank you for using RVM!
# I sincerely hope that RVM helps to make your life easier and more enjoyable!
!!
#
# ~Wayne

admin@ADMIN-501MOKPBO /usr/local/rvm
$
```

To make the rvm installation effective, you move to your home (login) directory, and then edit the following file,

.bashrc

If you can use either vi(vim) or emacs, it would be enough.

Add the next line to .bashrc, at the last of the file,

```
[[ -s $HOME/.rvm/scripts/rvm ]] && source "$HOME/.rvm/scripts/rvm"
```

When you edit with vim, moving cursor by pressing arrow keys, to the bottom line, and then, type 'o' to 'insert the new line to the next line.'

If you finished typing the entire line, the press [ESC] to switch to 'cursor movement mode,' and then type ':w' to submit command of 'overwrite file,' and type ':q' to exit from vim.

Please consult to the internet environment regarding "How to use 'vim'."

When you edit with emacs, you can directly input the command.

By pressing Ctrl-x, Ctrl-s to overwrite file, and press Ctrl-x, Ctrl-c to exit from emacs. There are not 'insert(input) mode,' and 'command mode' in emacs. Also please consult to the internet environment regarding "How to use 'emacs.'"

By typing the command.

```
pwd(print working directory)
```

to check where you are, If you are at the home directory, then type the following command to see the content of .bashrc file.

```
cat .bashrc
```



```
admin@ADMIN-501MOKPBO
$ pwd
/home/admin
admin@ADMIN-501MOKPBO ~
$ cat .bashrc
PATH=$PATH:$HOME/.rvm/bin # Add RVM to PATH for scripting
[[ -s $HOME/.rvm/scripts/rvm ]] && source "$HOME/.rvm/scripts/rvm"
admin@ADMIN-501MOKPBO ~
$ source ~/.bashrc
```

If you want to reflect the content of .bashrc to the shell, then type ,

```
source ~/.bashrc
```

Futher more, add the following one line to the file “.bash_profile”

```
export LIBRARY_PATH=/usr/lib/w32api:$LD_LIBRARY_PATH
```

A terminal window with a black background and white text. The prompt is 'admin@ADMIN-501MOKPB0 ~'. The user has entered '\$ cat .bash_profile'. The output shows the following lines: 'export LIBRARY_PATH=/usr/lib/w32api:\$LD_LIBRARY_PATH', '[[-s \$HOME/.rvm/scripts/rvm]] && source "\$HOME/.rvm/scripts/rvm"', and 'PATH=\$PATH:\$HOME/.rvm/bin # Add RVM to PATH for scripting'.

```
admin@ADMIN-501MOKPB0 ~  
$ cat .bash_profile  
  
export LIBRARY_PATH=/usr/lib/w32api:$LD_LIBRARY_PATH  
[[ -s $HOME/.rvm/scripts/rvm ]] && source "$HOME/.rvm/scripts/rvm"  
PATH=$PATH:$HOME/.rvm/bin # Add RVM to PATH for scripting
```

In above sample, I just added the same modification with “.bashrc” to the file “.bash_profile.”

If you finish running git-bash, and then start again, then you would be able to use rvm. Just make sure that the version will be replied with the following command,

```
rvm --version
```

Step 8: Install pik

In the Windows 7 non-Cygwin environment, (it seems) rvm does not work fine, so we try to use pik. Please type the following command for gem,

```
gem install pik
```

If you see the message ‘successfully installed pik-x.x....,’ then the installation was successful.

```

tkuo@DELL-E5500-KOBA /usr/local/rvm (master)
$ gem install pik
Fetching: pik-0.2.8.gem (100%)
-----
* If you're upgrading from a version <= 0.1.1, you'll want to delete the pik.bat
t file
  from all of your ruby versions. Gem uninstall should do the trick.
* Install pik to a location that's in your path, but someplace other than your
ruby\bin dir
  If you're upgrading from a more recent version, pik_install will overwrite th
e older files as needed.

>path
  PATH=C:\tools\;C:\ruby\186-p368-mingw32\bin;C:\WINDOWS\system32;C:\WINDOWS

>pik_install C:\tools

* If this is a first-time install, add all the versions of ruby that you want t
o use with pik

>pik add
Adding: 186: ruby 1.8.6 (2009-03-31 patchlevel 368) [i386-mingw32]
  Located at: c:/ruby/186-p368-mingw32/bin

>pik add C:\ruby\IronRuby-091\bin
Adding: 091: IronRuby 0.9.1.0 on .NET 2.0.0.0
  Located at: C:/ruby/IronRuby-091/bin

>pik add C:\ruby\jruby-1.4.0RC1\bin
Adding: 140: jruby 1.4.0RC1 (ruby 1.8.7 patchlevel 174) (2009-09-30 80c263b
) (Java HotSpot(TM) Client VM 1.6.0_14) [x86-java]
  Located at: C:/ruby/jruby-1.4.0RC1/bin

-----

Successfully installed pik-0.2.8
1 gem installed
Installing ri documentation for pik-0.2.8...
Installing RDoc documentation for pik-0.2.8...

tkuo@DELL-E5500-KOBA /usr/local/rvm (master)

```

Then, we make sure that the “c:\Ruby192\bin” path is added to the environment variable, by typing

```
echo $PATH
```

```

tkuo@DELL-E5500-KOBA /usr/local/rvm (master)
$ echo $PATH
/c/Users/Ikuo/bin:./usr/local/bin:/mingw/bin:/bin:/c/PROGRA~1/Borland/CBUILD~1/
Projects/Bpl:/c/PROGRA~1/Borland/CBUILD~1/Bin:/c/Program Files/Java/jdk1.6.0_18/
bin:/c/windows/system32:/c/windows:/c/windows/System32/wbem:/c/windows/System32/
WindowsPowerShell/v1.0:/c/Program Files/WIDCOMM/Bluetooth Software:/c/Program
Files/NTRU Cryptosystems/NTRU TCG Software Stack/bin:/c/Program Files/Wave Syst
ems Corp/Gemalto/Access Client/v5:/c/Program Files/Windows Live/Shared:/c/Progr
am Files/MySQL/MySQL Server 5.0/bin:/c/Ruby192/bin

```

Now we move to the directory c:\Ruby192\bin and then type the following command,

```
pik_install c:/ruby192/bin
```

(Please note, that we type c:/(slash) not \ (back slash).

```

Ikuo@DELL-E5500-KOBA /c/ruby192/bin
$ pik_install c:/ruby192/bin
Thank you for using pik.

mkdir -p c:/ruby192/bin
mkdir -p C:\Users\Ikuo\.pik
Installing to c:/ruby192/bin

cp c:/Ruby192/lib/ruby/gems/1.9.1/gems/pik-0.2.8/tools/pik_runner.exe c:/ruby192/bin
cp c:/Ruby192/lib/ruby/gems/1.9.1/gems/pik-0.2.8/tools/pik.bat c:/ruby192/bin
cp c:/Ruby192/lib/ruby/gems/1.9.1/gems/pik-0.2.8/tools/pik.ps1 c:/ruby192/bin

creating C:\Users\Ikuo\.pik\.pikrc

pik is installed

if you want to use pik with git bash, add the following line to your ~/.bashrc:

[[ -s $USERPROFILE/.pik/.pikrc ]] && source $USERPROFILE/.pik/.pikrc

Ikuo@DELL-E5500-KOBA /c/ruby192/bin

```

Now we go back to the home(~) directory to edit .bashrc file. Please follow the instruction at the last two lines from pik_install.

As for how to edit .bashrc with vi(vim) or emacs, please refer to the explanation shown in the step 7.

```

Ikuo@DELL-E5500-KOBA /c/ruby192/bin
$ cd

Ikuo@DELL-E5500-KOBA ~
$ vim .bashrc

Ikuo@DELL-E5500-KOBA ~
$ pwd
/c/Users/Ikuo

Ikuo@DELL-E5500-KOBA ~
$ cat .bashrc
[[ -s $USERPROFILE/.pik/.pikrc ]] && source $USERPROFILE/.pik/.pikrc

Ikuo@DELL-E5500-KOBA ~
$

```

Just type the following:

```
pik list
```

```

Ikuo@DELL-E5500-KOBA ~
$ pik list
* 192: ruby 1.9.2p290 (2011-07-09) [i386-mingw32]

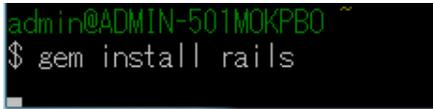
```

If you see the above message, the pik installation was successful.

Step 9: Install rails

Type the following command,

```
gem install rails
```



```
admin@ADMIN-501MOKPBO ~  
$ gem install rails
```

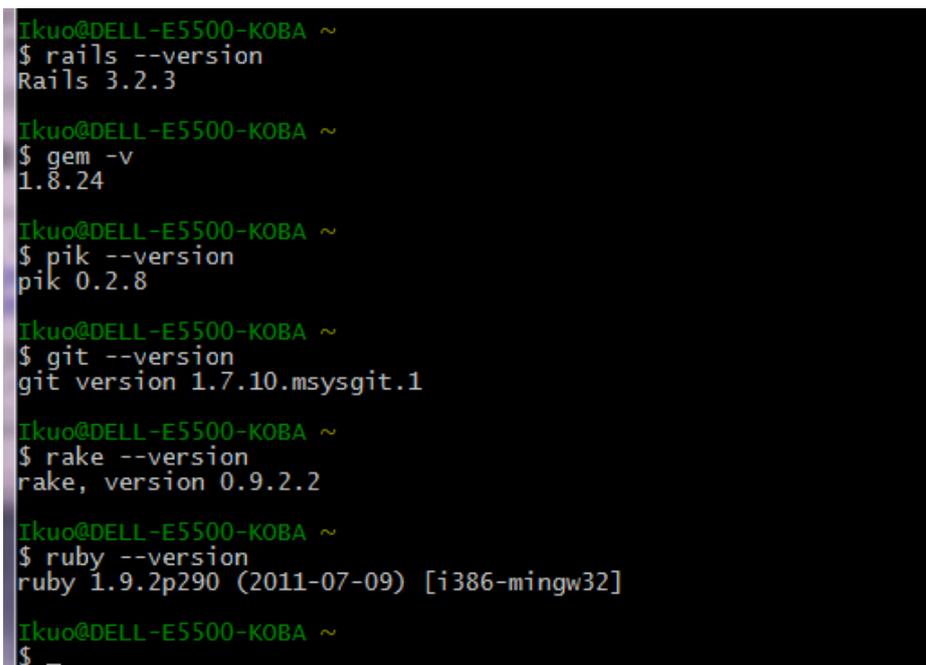
It will complain that it cannot convert UTF-8 characters into Windows-31J environment, but just ignore its complaint. You can read English message.

After the installation, type the following to make sure it replies the version number.

```
rails --version
```

Step 10: Confirmation of Version Numbers

With the above installation procedures, if you get the following version numbers, the installation was successful.



```
Ikuro@DELL-E5500-KOBA ~  
$ rails --version  
Rails 3.2.3  
  
Ikuro@DELL-E5500-KOBA ~  
$ gem -v  
1.8.24  
  
Ikuro@DELL-E5500-KOBA ~  
$ pik --version  
pik 0.2.8  
  
Ikuro@DELL-E5500-KOBA ~  
$ git --version  
git version 1.7.10.msysgit.1  
  
Ikuro@DELL-E5500-KOBA ~  
$ rake --version  
rake, version 0.9.2.2  
  
Ikuro@DELL-E5500-KOBA ~  
$ ruby --version  
ruby 1.9.2p290 (2011-07-09) [i386-mingw32]  
  
Ikuro@DELL-E5500-KOBA ~  
$
```

You may get the newer version, but it will be okay. It is quite likely if you follow this instruction several weeks(even days) after the dates of my screen shots.

All done! Then, enjoy programming with Ruby on Rails together!