Rally Ruby REST API Configuration Guide and parent_portfolio_items.rb Script Usage

Introduction

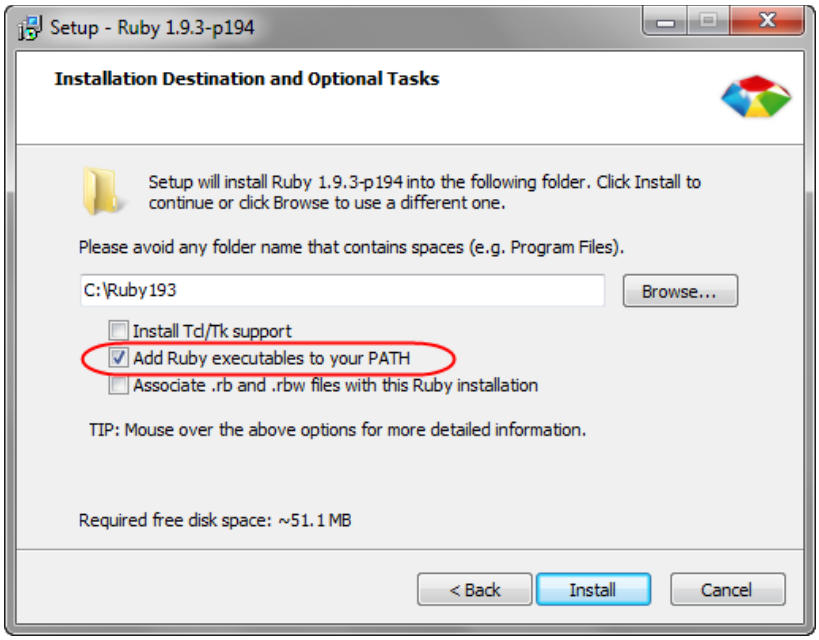
The Rally REST API tool is built on a Ruby interface to the Rally REST web service API. This script is *not officially supported* and is *used at your own risk.*

This document is composed of the following sections:

1. Installing Ruby on Windows

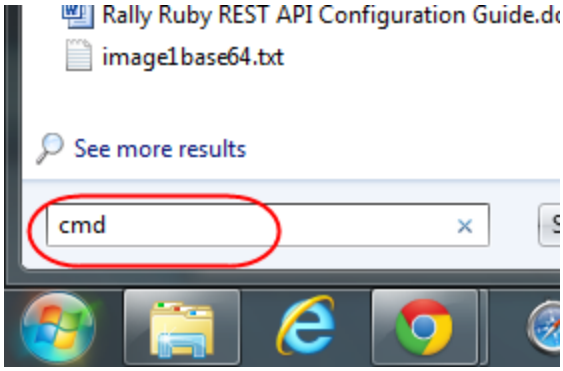
Install the Ruby 1.9.3 (preferable) Runtime Environment: http://rubyinstaller.org/downloads/

1. During installation, please make sure to add the Ruby executable to your Path:

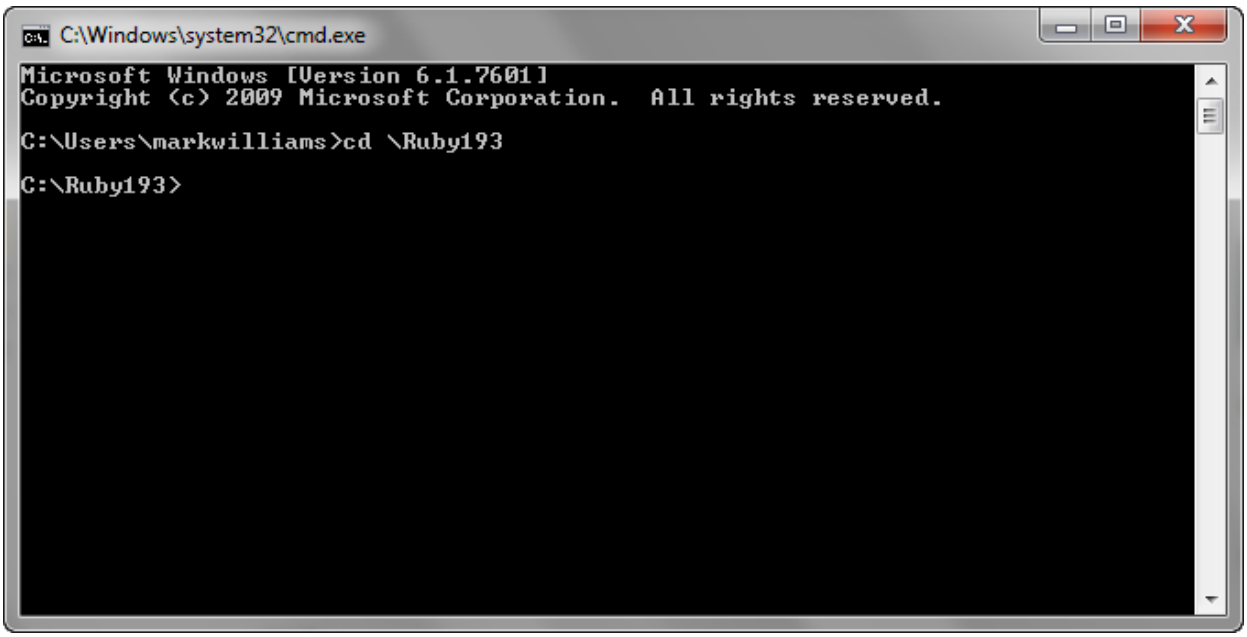


2. Open a command prompt window and go to the ruby directory that was created. In this example, Ruby was installed into C:\Ruby193:

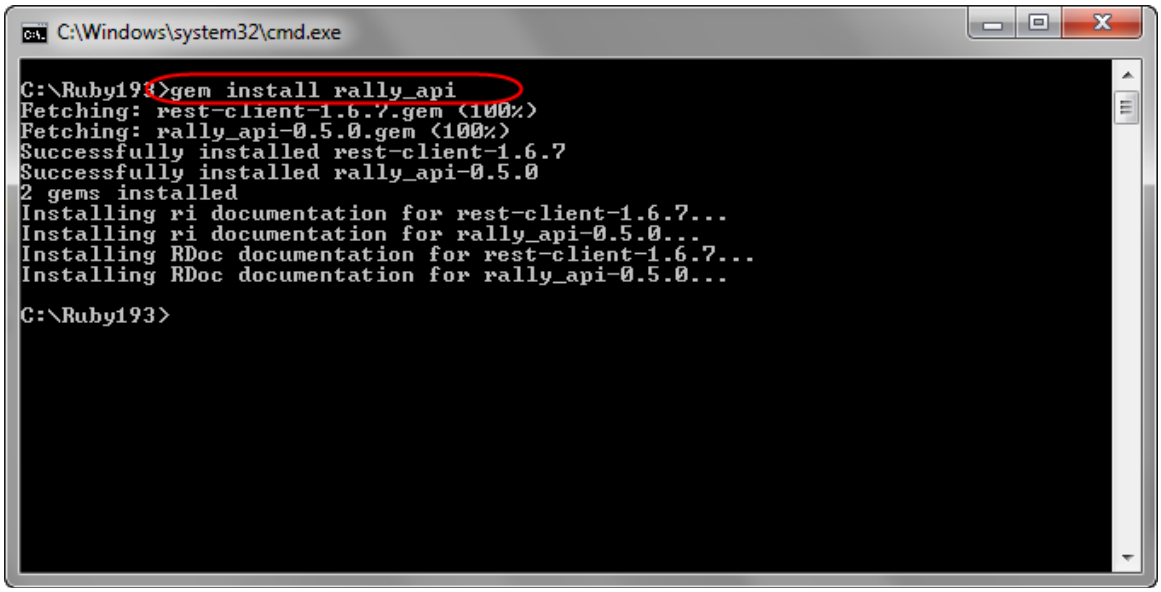
    a. Click on your “Start” button, then enter cmd into the search dialog and hit Enter.



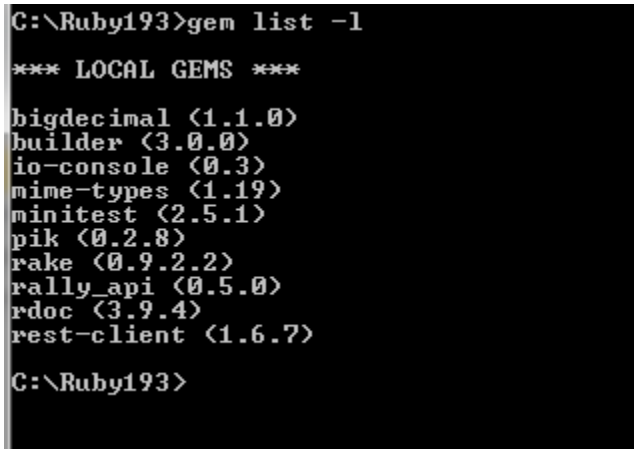
3. The Command prompt window appears. Navigate to where you installed Ruby:



4. Install the rally_api gem. This will also install its dependent gems.



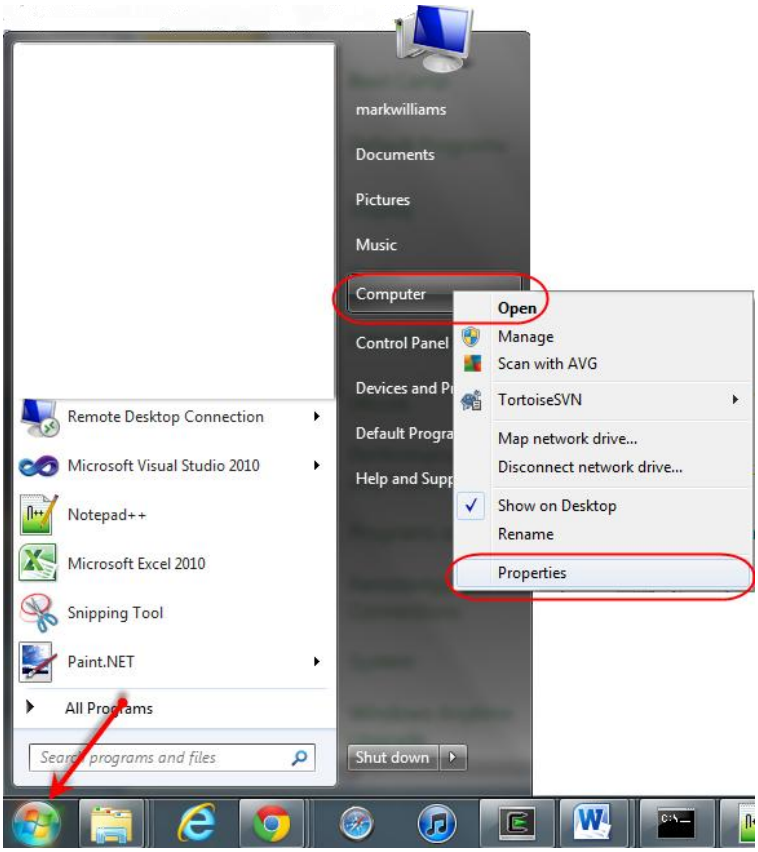
5. When finished, you can verify all RubyGems installed by typing "**`gem list -l`**":



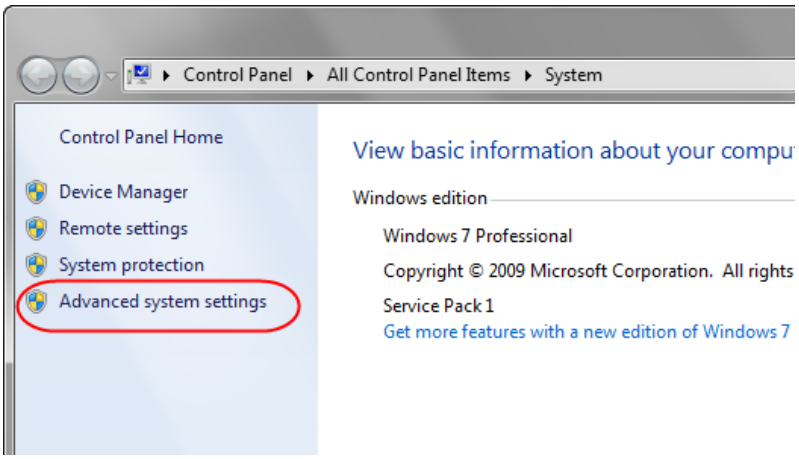
2. Configuring a Proxy Server (If Your Company Uses a Proxy)

1. If your company is behind a firewall or a proxy server, you may need to take additional steps in order to run Rally ruby scripting tools. To access the internet via a proxy-server using Windows, go to:

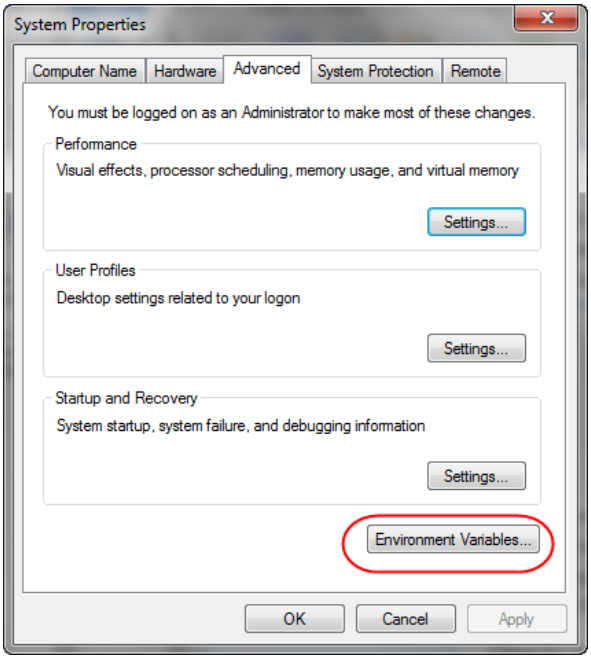
   Start -> Computer (Right Click) -> Properties:



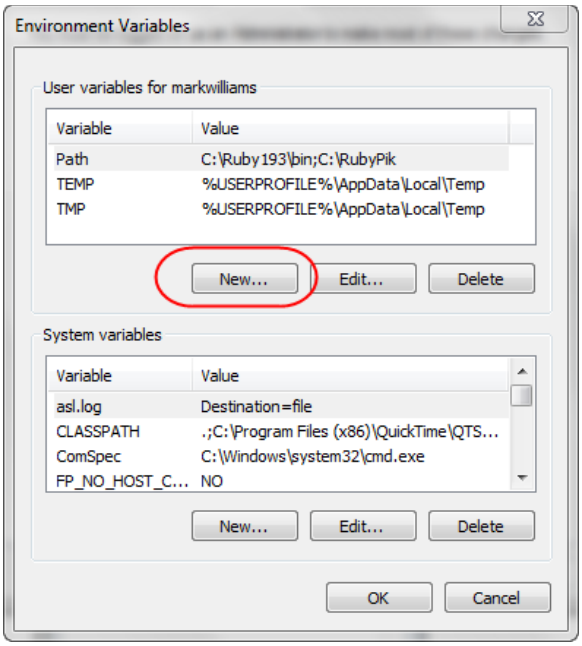
2. Advanced System Settings:



3. Environment Variables:



4. Use the New button to create a new environment variable:

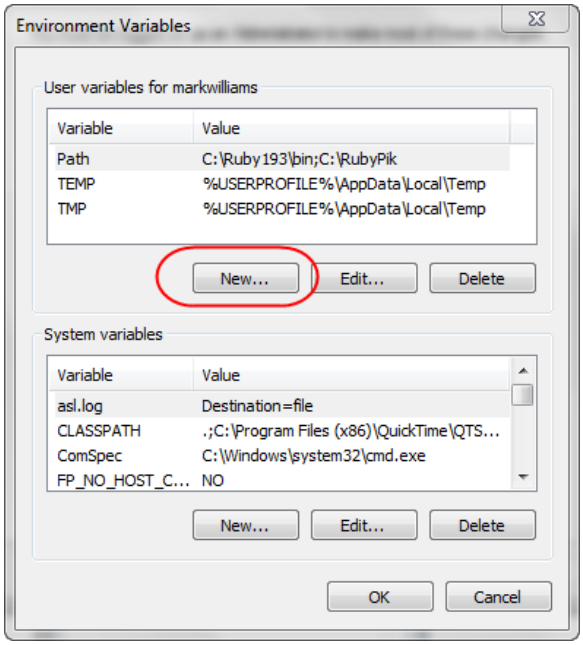


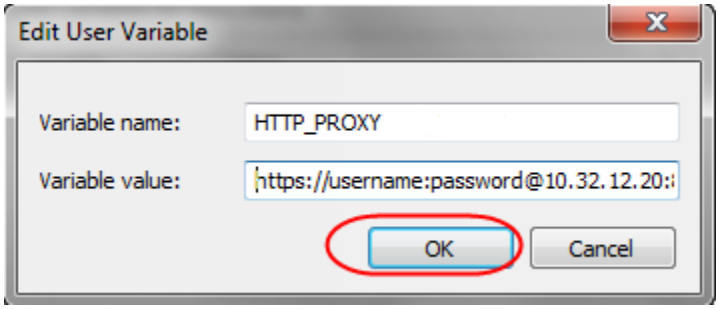
5. Create the following environment variables:

- **HTTP_PROXY**
- **HTTPS_PROXY**
- **FTP_PROXY**

The value for each of the 3 variables is *usually* the same and of the general format:
http://[name:password@]ipaddress:port/

6. Let's create HTTP_PROXY as an example:



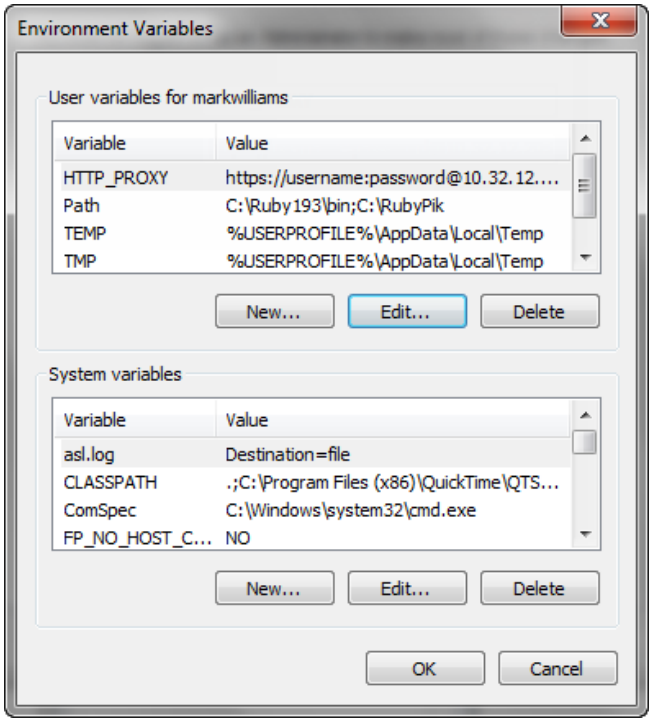


In this example we entered:

Variable name: HTTP_PROXY
Variable value: https://username:password@10.32.12.20:8080

The actual values of username, password, and the proxy server address:port (10.32.12.20:8080) are going to vary according to your environment. You may need to check with your IT department concerning the appropriate information.

Completed Environment Variable Entry:



7. Please create ***both*** **HTTP_PROXY** and **HTTPS_PROXY** variables

8. Note that you will have to open a ***New Command Prompt window*** after creating any environment variables in order for them to take effect in the Command prompt.

3. Configuring and Using the Parent Portfolio Items Script

1. Create directory for script and associated files:

   C:\Users\username\Documents\Rally Parent Portfolio Items\

2. Download the parent_portfolio_items.rb script and the my_vars.rb file to the above directory

3. Using a text editor, customize the code parameters in the my_vars.rb file for your environment.

```
my_vars.rb:
===========

# Connection Parameters
$my_username                = 'user@company.com'
$my_password                = 'password'
$my_base_url                = "https://rally1.rallydev.com/slm"
$my_workspace               = "My Workspace"
$my_project                 = "My Project"
$wsapi_version              = "1.40"
$filename                   = "parent_portfolio_items.csv"
```

4. Create a CSV file called parent_portfolio_items.csv, located in the same directory as the Ruby script, that contains the Portfolio Items that you want to import, in comma-separated format. For example:

```
Child Formatted ID, Child PI Type, Child PI Name, Parent Formatted ID, Parent PI
Type, Parent PI Name
F469, Feature, Feature Test01, I42, Initiative, Initiative Test01
F470, Feature, Feature Test02, I43, Initiative, Initiative Test02
F471, Feature, Feature Test03, I44, Initiative, Initiative Test03
F472, Feature, Feature Test04, I45, Initiative, Initiative Test04
F473, Feature, Feature Test05, I46, Initiative, Initiative Test05
F474, Feature, Feature Test06, I47, Initiative, Initiative Test06
F475, Feature ,Feature Test07, I48, Initiative, Initiative Test07
```

Important: The script does not check to see if Parent is a valid assignment for the Portfolio Item of concern – i.e. it isn't as robust as to check to see whether or not you're trying to assign a Parent to a top-level-of-the-hierarchy Portfolio Item. If this assignment is attempted, the script with throw an exception, but will continue processing remaining rows.

5. Run the script.

```
      C:\> ruby parent_portfolio_items.rb
Child PI Item F469: Feature Test01 successfully updated parent to:
   ==> Parent P3-42: Initiative Test01
Child PI Item F470: Feature Test02 successfully updated parent to:
   ==> Parent P3-43: Initiative Test02
Child PI Item F471: Feature Test03 successfully updated parent to:
   ==> Parent P3-44: Initiative Test03
Child PI Item F472: Feature Test04 successfully updated parent to:
   ==> Parent P3-45: Initiative Test04
Child PI Item F473: Feature Test05 successfully updated parent to:
   ==> Parent P3-46: Initiative Test05
Child PI Item F474: Feature Test06 successfully updated parent to:
   ==> Parent P3-47: Initiative Test06
Child PI Item F475: Feature Test07 successfully updated parent to:
```

```
==> Parent P3-48: Initiative Test07
```

Warning!!: This script will ***replace/over-write all*** parent assignments for Children specified in the Child FormattedID column. Make sure you have the correct data in this column before running!