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

Introduction

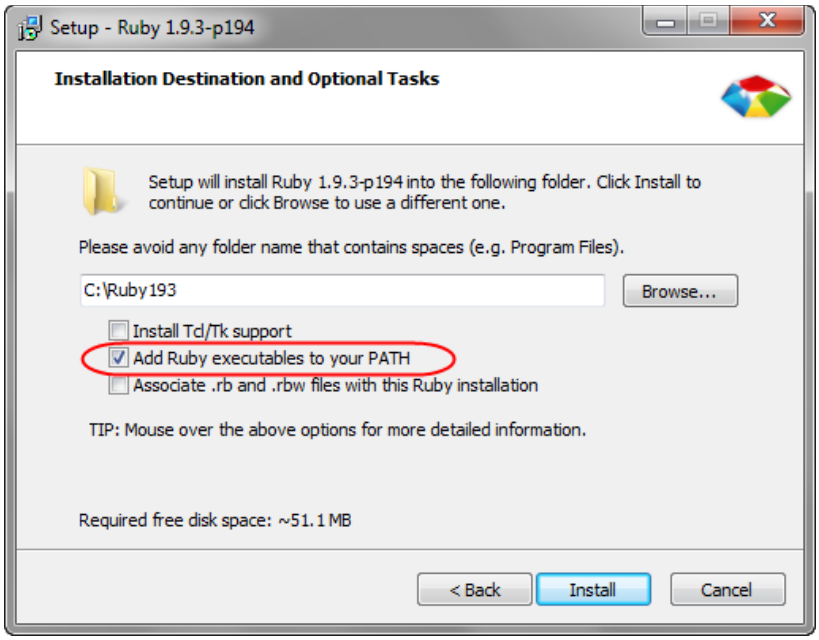
The Rally REST API tool is built on a Ruby interface to the Rally REST web service API. This script is *not officially supported* and is *used at your own risk.*

This document is composed of the following sections:

1. Installing Ruby on Windows

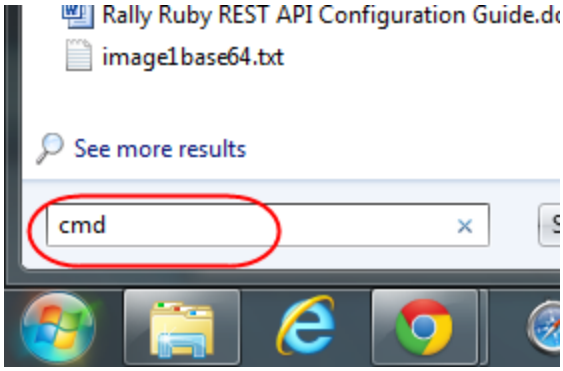
Install the Ruby 1.9.3 (preferable) Runtime Environment: http://rubyinstaller.org/downloads/

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

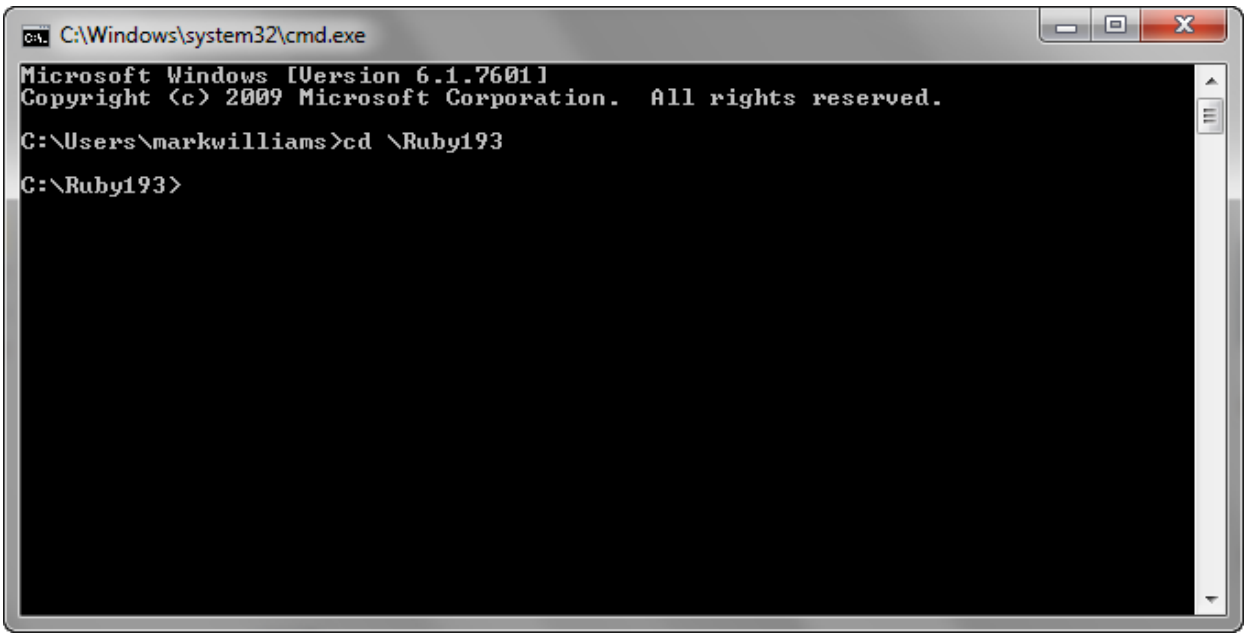


2. Open a command prompt window and go to the ruby directory that was created. In this example, Ruby was installed into C:\Ruby193:

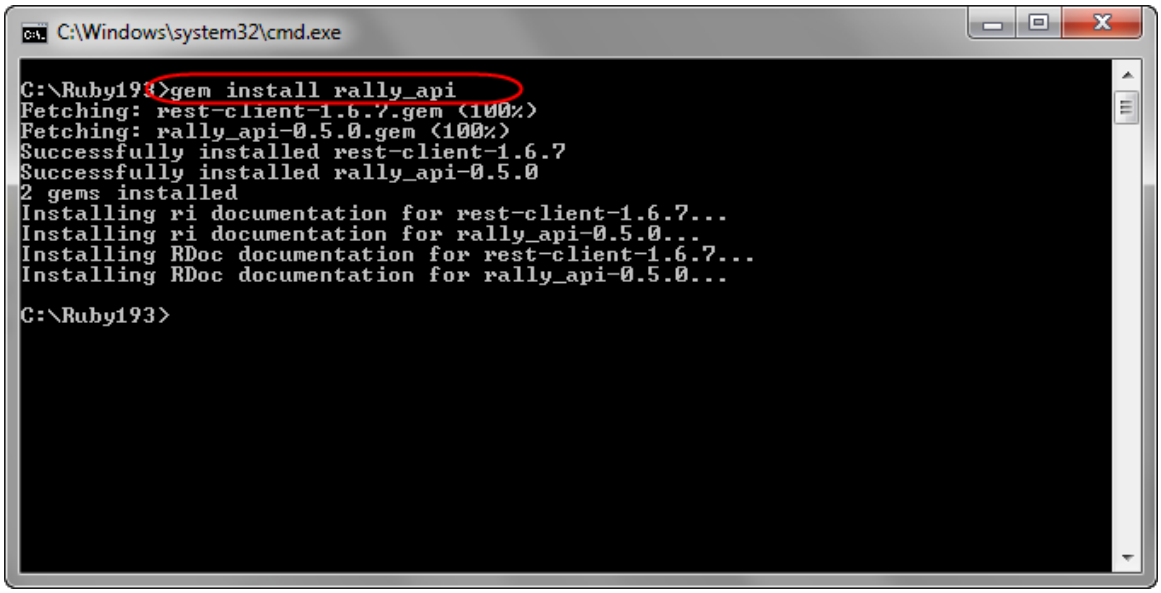
    a. Click on your “Start” button, then enter cmd into the search dialog and hit Enter.



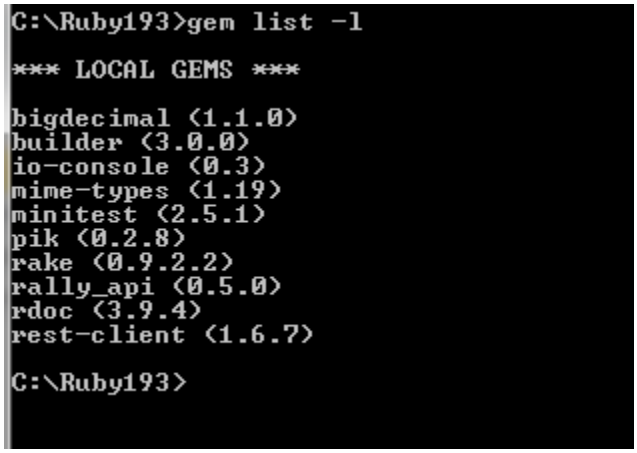
3. The Command prompt window appears. Navigate to where you installed Ruby:



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



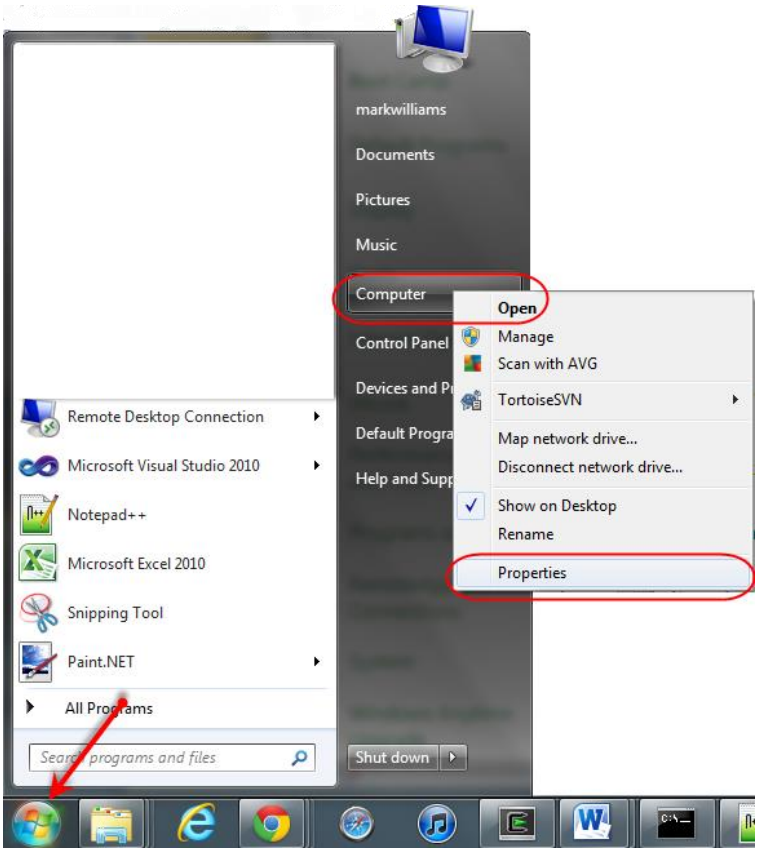
5. When finished, you can verify all RubyGems installed by typing "**`gem list -l`**":



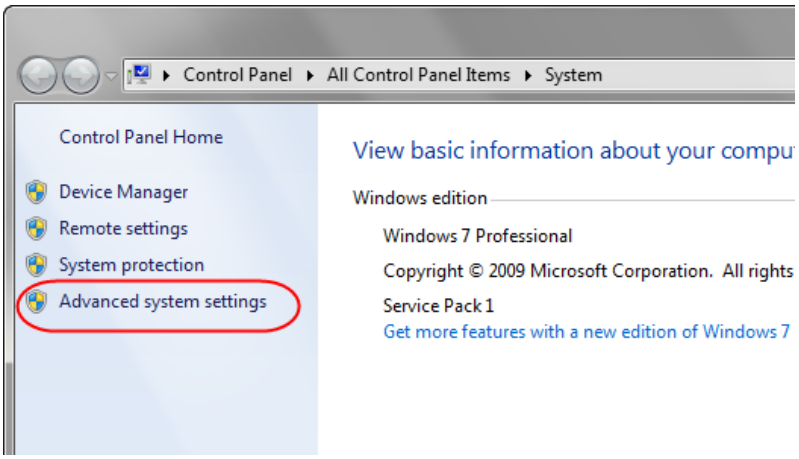
2. Configuring a Proxy Server

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:

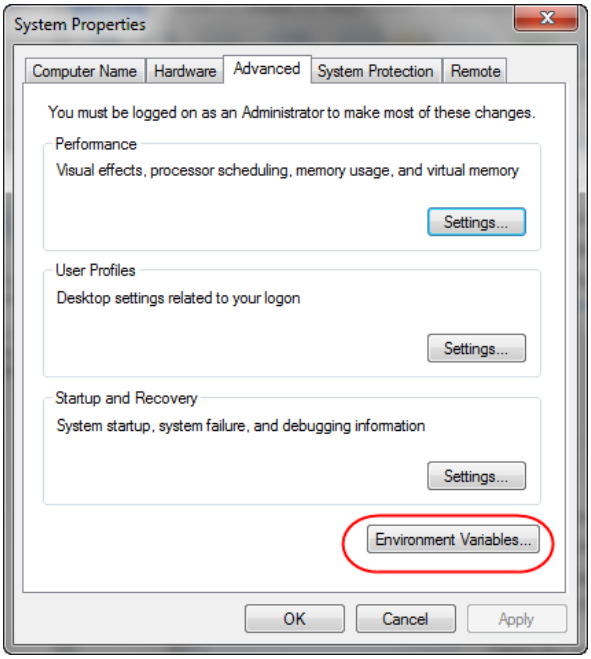
   Start -> Computer (Right Click) -> Properties:



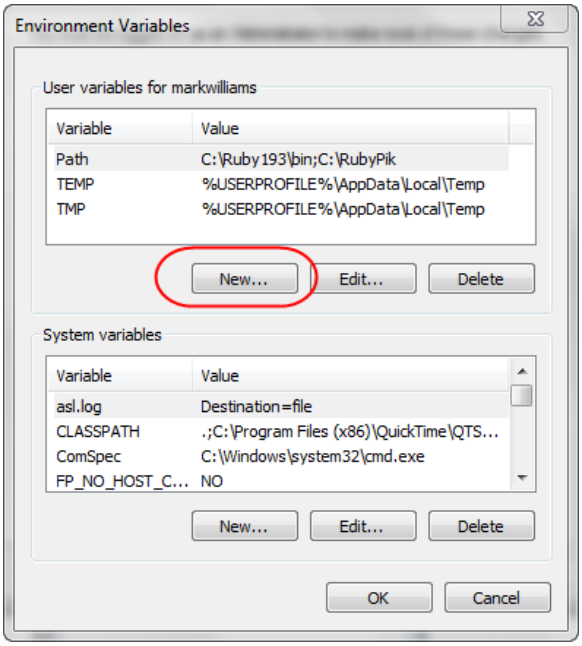
2. Advanced System Settings:



3. Environment Variables:



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

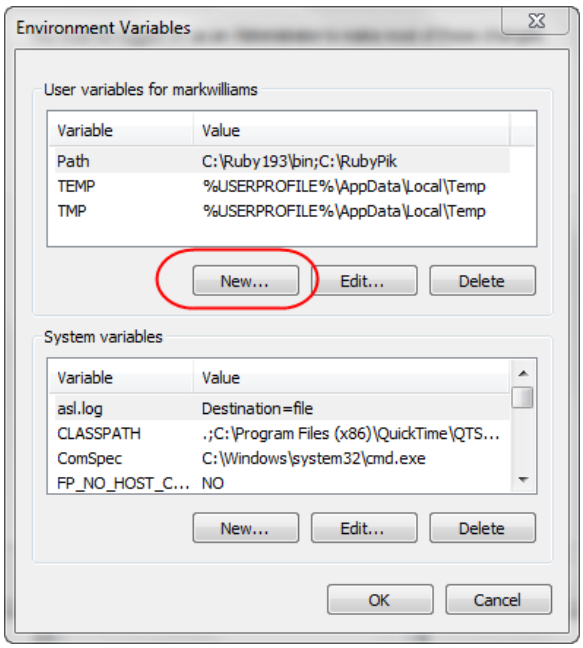


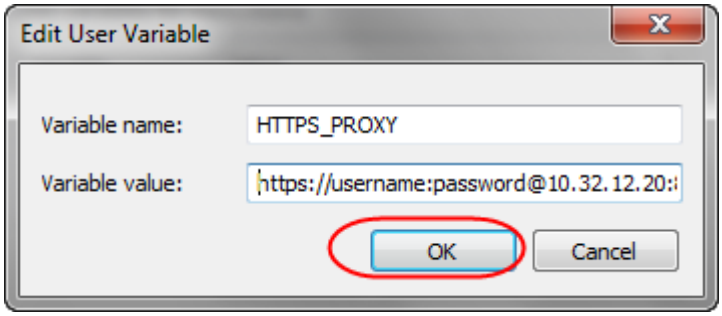
5. Create the following environment variables:

- **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 HTTPS_PROXY as an example:



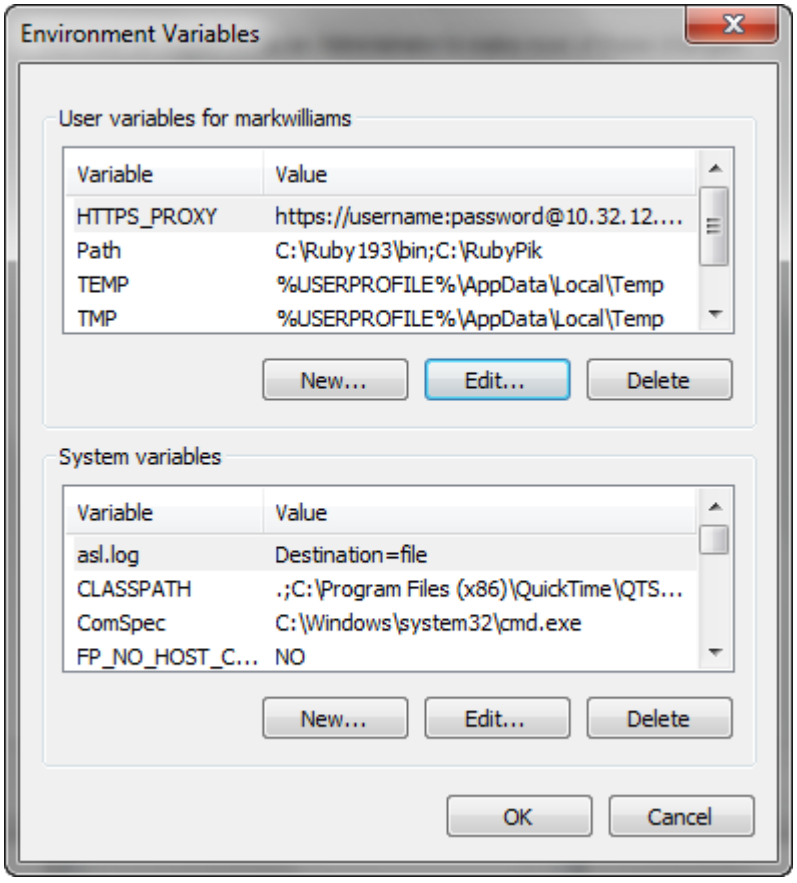


In this example we entered:

Variable name: HTTPS_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. 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 Export Test Set Script

1. Create directory for script and associated files:

   C:\Users\username\Documents\Rally Export Test Set\

2. Download the export_test_set.rb and the my_vars.rb script bundle from Github repository to the above directory

3. Using a text editor, customize the code parameters in the my_vars.rb file for your environment. ***Note:*** The script defaults to using *tab-delimited* output. You can change this my adjusting the following variable:

```
$my_delim                = "\t"

TO:

$my_delim                = ","

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

# Connection Parameters
$my_base_url             = "https://rally1.rallydev.com/slm"
$my_username             = "user@company.com"
$my_password             = "topsecret"
$my_page_size            = 200
$my_limit                = 50000
$my_delim                = "\t"
$wsapi_version           = "1.43"

# Workspace/project info
$my_workspace            = "My Workspace"
$my_project              = "My Project"

# Test Set Info
$my_test_set_formatted_id = "TS7"

# output
$my_output_file          = "testset.txt"
```

4. Run the script.

```
C:\> ruby export_test_set.rb
Connecting to Rally.
Querying Rally for Test Set.
Fount Test Set: TS7. Looking for Test Cases.
Exporting test cases to file: testset.txt.
Total Test Cases to Export: 4
Test Case TC2: 1 of 4 exported.
Test Case TC3: 2 of 4 exported.
Test Case TC4: 3 of 4 exported.
Test Case TC5: 4 of 4 exported.
Done! Test Cases for Test Set: TS7 written to: testset.txt.
```

Note: If you require import of fields/data that differ from the template in this example, you'll need to customize the code to suit your needs.