

## ISO Developer's Toolkit - Installing on Windows

2015 CDI Workshop

May 11, 2015 Stan Smith, USGS

## Do I need to install Ruby?



- I will need to install Ruby if ...
  - o ... I want to participate in extending the ISO Metadata Toolkit
  - o ... I want to customize the mdTranslator for my own purpose
  - ... I want to integrate the adiwg-mdTranslator gem into local systems (either by code or CLI)
  - o ... I want to host the mdTranslator as a local web service
- I do not need to install Ruby if ...
  - o ... I will use the publicly hosted mdTranslator API



## Why Ruby?



- Ruby is a dynamic, Object Oriented, cross-platform, opensource, general-purpose programming language written in C (not to be confused with 'Ruby on Rails')
- Initial development in mid 1990s by Yukihiro Matsumoto
- Gained wide acceptance and popularity
- Ruby has more than 100,000 Gems available for specialized tasks
- Interpretive language
- The MinGW compiler is used for Windows installations
- Ruby project on GitHub <a href="https://github.com/ruby">https://github.com/ruby</a>

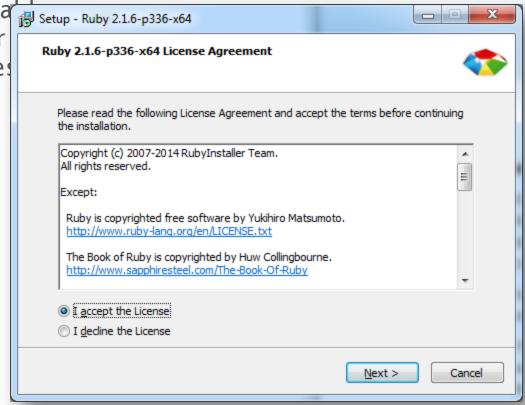




Download the appropriate Windows installer from http://rubyinstaller.org/. In this example we use Ruby 2.1.6 x64.

Version for 2.2.2 is availa Setup - Ruby 2.1.6-p336-x64 but I like to give time for RubyGem providers to tes against new versions.

- Navigate to the download directory and double-click the self-extracting 7z file.
- Accept the license ...



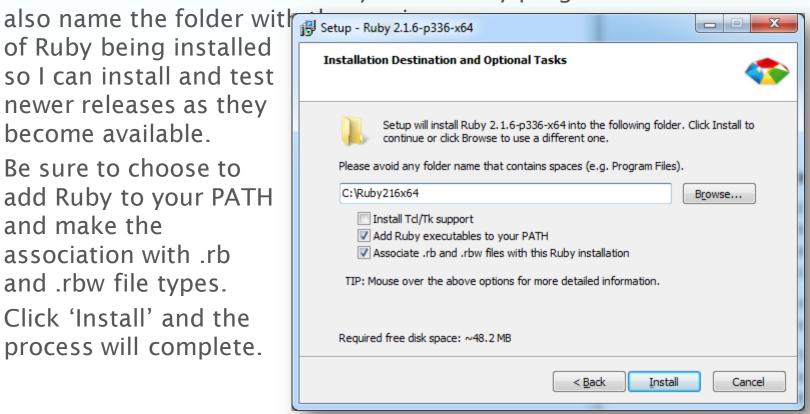




4. Provide a location for the installation. I use the root of the C:\ drive to be sure the binaries are easy for all Ruby programs to find. I

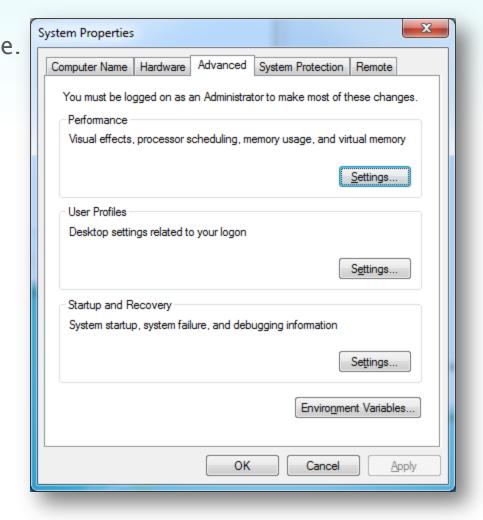
of Ruby being installed so I can install and test newer releases as they become available.

- 5. Be sure to choose to add Ruby to your PATH and make the association with .rb and .rbw file types.
- 6. Click 'Install' and the process will complete.





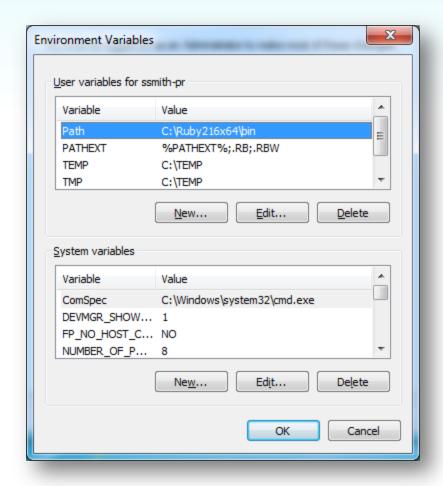
7. The installations is complete. So let's verify it. Check that the path was added to the Windows **Environment Variables.** The PATH can be viewed and edited from the 'Advanced' tab of 'System Properties'. Click the 'Environment Variables ...' button.







8. Be sure you see the path to the folder you installed Ruby in.





- With Ruby installed, test that Ruby works properly in your environment. Start a command window (launch cmd.exe from the start menu).
- 10. From the prompt check the version of Ruby installed: > ruby -v

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\ssmith-pr>ruby -v
ruby 2.1.6p336 (2015-04-13 revision 50298) [x64-mingw32]
C:\Users\ssmith-pr>
```



- 11. You can also start the 'Interactive Ruby' console and write a few lines of Ruby code just to see that all works properly.
  - > irb
  - > a = 'Hello'
  - > b = 'World!'
  - > puts a + b Hello World!

```
C:\Windows\system32\cmd.exe
   Users\ssmith-pr>irb
    (main):001:0> a = 'Hello '
 b(main):002:0> b = 'World!'
 rb(main):003:0> puts a + b
⇒ nil
irb(main):004:0> exit
C:\Users\ssmith-pr>
```



12. Ruby gets much of its power from the rich repository of RubyGems or code libraries. These are written to easily plug into and be consumed by Ruby programs.

Check the gems installed with the Ruby installation:

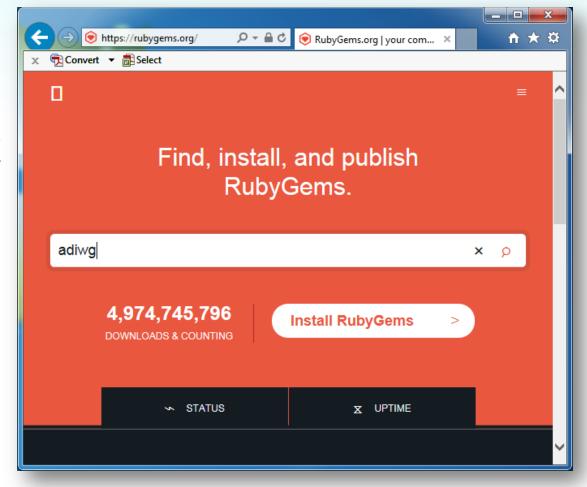
> gem list

13. Note that the RubyGem 'gem' was also installed although not listed.

```
C:\Windows\system32\cmd.exe
C:\Users\ssmith-pr>gem list
*** LOCAL GEMS ***
bigdecimal (1.2.4)
test-unit (2.1.6.0)
C:\Users\ssmith-pr>
```

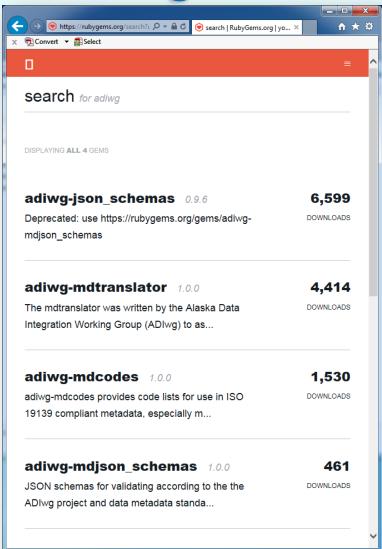


14. Thousands of
RubyGems are
available for
download at
<a href="http://rubygems.org/">http://rubygems.org/</a>
including those
written by ADIwg.
Enter 'adiwg' and
press 'Enter'.





- 15. The ADIwg gems available are listed along with their current download counts.
- 16. In this case we want to install the 'adiwg-mdtranslator' gem. From the command line type: > gem install adiwg-mdtranslator





17. The adiwg-mdtranslator gem will install along with all its other

RubyGem dependencies. Notice 6 gems were installed.

```
C:\Windows\system32\cmd.exe
   Users\ssmith-pr>gem install adiwg-mdtranslator
Fetching: builder-3.2.2.gem (100%)
Successfully installed builder-3.2.2
Fetching: thor-0.19.1.gem (100%)
Successfully installed thor-0.19.1
Fetching: uúidtools-2.1.5.gem (100%)
Successfully installed uuidtools-2.1.5
Fetching: json-schema-2.4.1.gem (100%)
Successfully installed json-schema-2.4.1
Fetching: adiwg-mdjson_schemas-1.0.0.gem (100%)
Successfully installed adiwg-mdjson_schemas-1.0.0
Fetching: adiwg-mdtranslator-1.0.0.gem (100%)
Successfully installed adiwg-mdtranslator-1.0.0
Parsing documentation for adiwg-mdjson_schemas-1.0.0
Installing ri documentation for adiwg-mdjson_schemas-1.0.0
Parsing documentation for adiwg-mdtranslator-1.0.0
Installing ri documentation for adiwg-mdtranslator-1.0.0
Parsing documentation for builder-3.2.2
Installing ri documentation for builder-3.2.2
Parsing documentation for json-schema-2.4.1
Installing ri documentation for json-schema-2.4.1
Parsing documentation for thor-0.19.1
Installing ri documentation for thor-0.19.1
Parsing documentation for uuidtools-2.1.5
Installing ri documentation for uuidtools-2.1.5
Done installing documentation for adiwg-mdjson_schemas, adiwg-mdtranslator, build
11 seconds
6 gems installed
C:\Users\ssmith-pr>
```



- 18. Since the adiwg-mdtranslator gem provides a Command Line Interface (CLI) we can check its installation by typing:
  - > mdtranslator help

```
C:\Windows\system32\cmd.exe
C:\Users\ssmith-pr>mdtranslator help
Commands:
  mdtranslator help [COMMAND] # Describe available commands or one specifi...
mdtranslator translate [FILE] # Pass JSON string or filename plus paramete...
                                           # Returns the version of mdTranslator
  mdtranslator version
 :\Users\ssmith-pr>
```



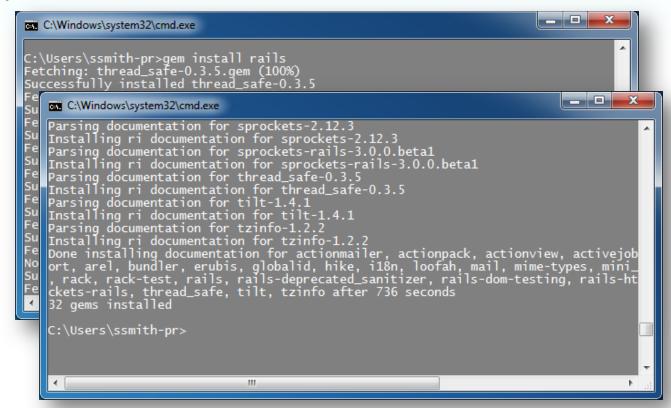


- 19. We can check the version and get more help on commands such as 'translate'.
- 20. Installation of Ruby and the mdTranslator are complete. We could use mdTranslator from the command line and pipe the result to a file.

```
C:\Windows\system32\cmd.exe
C:\Users\ssmith-pr>mdtranslator version
C:\Users\ssmith-pr>mdtranslator help translate
 mdtranslator translate [FILE]
Options:
  -r, [--reader=READER]
                                               # Name of reader to read your input
                                                 Default: mdJson
  -w, [--writer=WRITER]
                                                 Name of writer to create your meta
  -v. [--validate=VALIDATE]
                                                 Specify level of validation to be
                                                 Default: normal
                                                 Possible values: none, normal, str
Include tags for unused attributes
  -s, [--showAllTags], [--no-showAllTags]
                                                 On error réturn messages as format
  -m, [--messages=MESSAGES]
                                                 Default: text
                                                 Possible values: json, text
  -o, [--returnObject], [--no-returnObject]
                                                 Return full JSON object generated
Description:
  mdtranslator translate' provides command line access to the ADIWG metadata
  translator with options to select the input file reader, select writer output
  format, show empty tags in XML outputs, and choose level of validation for
  JSON inputs.
C:\Users\ssmith-pr>
```



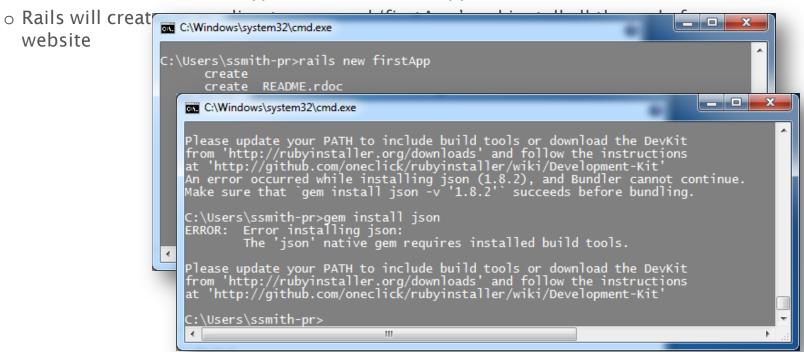
- 21. If we wish to use mdTranslator within a local web service we need to install Ruby on Rails (Rails).
- 22. With Ruby install. use Gem to do the work for us. > gem install rails
- 23. Notice Rails installed another 32 gems.





- 24. To test the Rails installation, ask rails to create a new website for US.
  - o Navigate to the folder to hold your new website.
  - o From the command line type: > rails new firstApp.

website





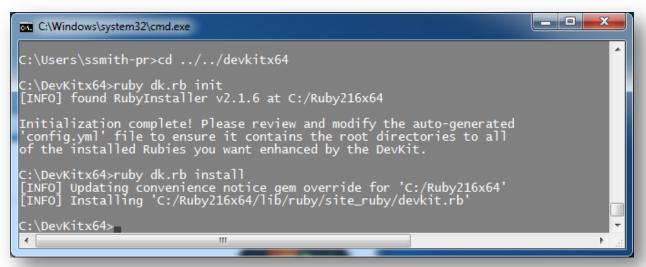
- 25. After many lines are reported back the web site failed to complete because the 'json' gem version 1.8.2 did not provided a version compiled for Windows. Unfortunately, glitches are common on Windows installations and the installation of DevKit will generally be required at some point. Best do it now!
- 26. Download the appropriate version of DevKit from <a href="http://rubyinstaller.org/downloads/">http://rubyinstaller.org/downloads/</a>.
- 27. I also install DevKit at the root of the C:\ directory.

  More information about installing DevKit can be found at

  <a href="https://github.com/OneClick/RubyInstaller/wiki/Development-Kit">https://github.com/OneClick/RubyInstaller/wiki/Development-Kit</a>.



- 28. To connect DevKit to the Ruby installation, navigate to the directory where DevKit installed.
- 29. From the command line type:
  - > ruby dk.rb init
- 30. Next type:
  - > ruby dk.rb install

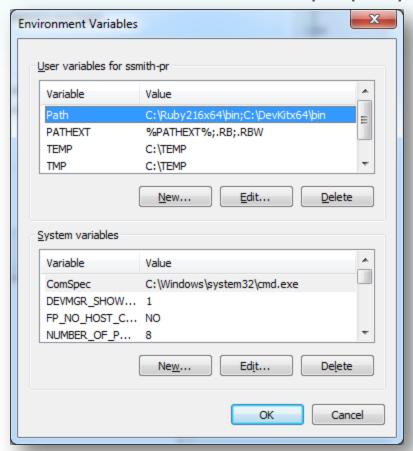






31. With DevKit installed, check that the PATH variable was properly

updated.





- 32. With the DevKit PATH now defined to Ruby, try for a properly installed version of the json gem. From the command line type: > gem install json
- 33. And this time it works!

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\ssmith-pr>gem install json
Temporarily enhancing PATH to include DevKit...
Building native extensions. This could take a while...
Successfully installed json-1.8.2
Parsing documentation for json-1.8.2
Installing ri documentation for json-1.8.2
Done installing documentation for json after 1 seconds
1 gem installed

C:\Users\ssmith-pr>
```



- 34. Try again to have Rails build a new web site. From the command line:
  - > rails new firstApp

35. After many lines you should soo a successful completion

```
C:\Windows\system32\cmd.exe
Installing sass-rails 5.0.3
Installing sdoc 0.4.1
Installing sqlite3 1.3.10
Installing turbolinks 2.5.3
Installing tzinfo-data 1.2015.4
Installing uglifier 2.7.1
Installing web-console 2.1.2
Bundle complete! 12 Gemfile dependencies, 54 gems now installed.
Use `bundle show [gemname]` to see where a bundled gem is installed.
Post-install message from rdoc:
Depending on your version of ruby, you may need to install ruby rdoc/ri da
<= 1.8.6 : unsupported
 = 1.8.7 : gem install rdoc-data; rdoc-data --install
 = 1.9.1 : gem install rdoc-data; rdoc-data --install
>= 1.9.2 : nothing to do! Yay!
C:\Users\ssmith-pr>
```



36. To test our new web site, navigate to its directory and start the rails server. From the command line type:

> rails server

37. Rails will start its default web server 'WEBrick' on localhost port 3000.

```
C:\Users\ssmith-pr>cd firstApp

C:\Users\ssmith-pr\firstApp>rails server

> Booting WEBrick

> Rails 4.2.1 application starting in development on http://localhost:3000

> Run `rails server -h` for more startup options

> Ctrl-C to shutdown server

[2015-04-27 10:50:35] INFO WEBrick 1.3.1

[2015-04-27 10:50:35] INFO ruby 2.1.6 (2015-04-13) [x64-mingw32]

[2015-04-27 10:50:35] INFO WEBrick::HTTPServer#start: pid=5404 port=3000

Started GET "/" for 127.0.0.1 at 2015-04-27 10:50:53 -0800

Processing by Rails::WelcomeController#index as HTML

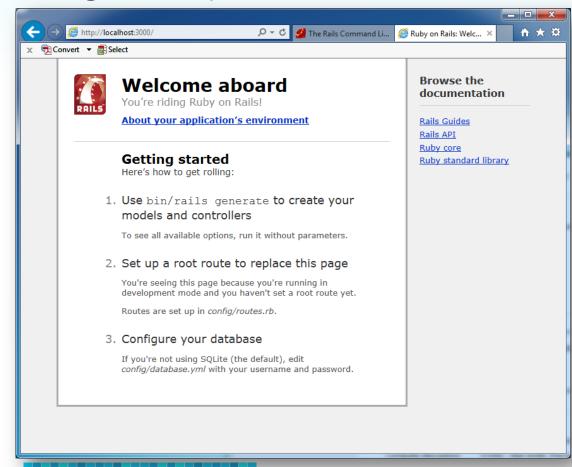
Rendered C:/Ruby216x64/lib/ruby/gems/2.1.0/gems/railties-4.2.1/lib/rails/t

)
Completed 200 OK in 37ms (Views: 19.5ms | ActiveRecord: 0.0ms)
```



38. Open you browser and navigate to <a href="http://localhost:3000">http://localhost:3000</a>.

You should see the Rails 'Welcome aboard' page.





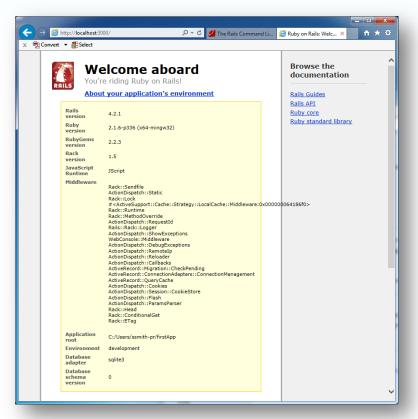


39. For more information about the installation click the 'About your application's environment' and Rails will display information about

the software and version used.

40. Time to start developing!

-- Almost --





41. Another common problem with Rails on Windows x64 is the gem 'tzinfo'. Windows does not include a time-zone table with the operating system so one needs to be downloaded. tzinfo will automatically set one up for Windows x32 but not x64. Find the 'Gemfile' line ...

gem 'tzinfo-data', platforms: [:mingw, :mswin]
edit to tell tzinfo to include support for Windows x64
gem 'tzinfo-data', platforms: [:mingw, :mswin, :x64\_mingw]

- 42. Run another gem installed with Rails, 'bundler'. Bundler tell Rails the gems and their versions to use with your Rails web site. From the terminal prompt to update the web site's gem files. Type: > bundle update.
- 43. Now you are ready. Really...

## ISO Developer's Toolkit - Linux Installation

2015 CDI Workshop

May 11, 2015 Presenter Here, USGS



## Ruby Version Manager



#### Options:

- o RVM: <a href="https://rvm.io/">https://rvm.io/</a>
- o rbenv: <a href="https://github.com/sstephenson/rbenv">https://github.com/sstephenson/rbenv</a>
- chruby: <a href="https://github.com/postmodern/chruby">https://github.com/postmodern/chruby</a>
- Easily switch between Ruby versions
- mdTranslator requires Ruby >= 2.1
- mdTranslator-rails requires 2.1.5



## Install mdTranslator (RVM)



```
#install rvm and ruby
```

- \$ gpg --keyserver hkp://pgp.mit.edu --recvkeys 409B6B1796C275462A1703113804BB82D39DC0E3
- \$ \curl -sSL https://get.rvm.io | bash -s
  stable -ruby
- \$ source ~/.rvm/scripts/rvm
- \$ gem install adiwg-mdtranslator
- \$ mdtranslator help translate



## Install mdTranslator-rails (RVM)

#install mdTranslator-rails
# if necessary, source rvm
\$ source ~/.rvm/scripts/rvm
\$ rvm install 2.1.5 #about 25MB download
\$ rvm use 2.1.5
\$ gem install bundler
\$ git clone https://github.com/adiwg/mdTranslator-rails.git #or git@github.com: adiwg/mdTranslator-rails.git
\$ cd mdTranslator-rails/

bundle update #will install rails

\$ rails server #or 'unicorn'

Alaska Data Integration working group

### **Questions?**



github.com/adiwg/mdWorkshop
www.adiwg.org/mdTools
mdtranslator.adiwg.org
mdbook.adiwg.org



# ISO Developer's Toolkit - CLI Integration

2015 CDI Workshop

May 11, 2015 Josh Bradley, USGS



#### mdTranslator CLI



- Command-line interface is provided with gem
- Supports all options available via API
- Allows simple integration with non-Ruby applications

#### mdTranslator CLI



#### > mdtranslator help translate

```
Usage:
 mdtranslator translate [FILE]
Options:
 -r, [--reader=READER]
                            # Name of reader to read your input
                            # Default: mdJson
 -w, [--writer=WRITER]
                            # Name of writer to create your metadata, or leave blank to validate input only
                            # Specify level of validation to be performed
 -v, [--validate=VALIDATE]
                            # Default: normal
                            # Possible values: none, normal, strict
 -s, [--showAllTags], [--no-showAllTags] # Include tags for unused attributes
 -m, [--messages=MESSAGES]
                                    # On error return messages as formatted text or ison object
                                    # Default: text
                                    # Possible values: json, text
 -o, [--returnObject], [--no-returnObject] # Return full JSON object generated by translator
```

## CLI: -o, [--returnObject]



> mdtranslator translate -o -w html file.json

```
"readerFormat": "json",
"readerStructurePass": true,
"readerStructureMessages": [],
"readerRequested": "mdJson",
"readerFound": "mdJson",
"readerVersionFound": "1.0.0",
"readerVersionUsed": "1.0",
"readerValidationLevel": "normal",
"readerValidationPass": true,
"readerValidationMessages" : [],
"readerExecutionPass": true,
"readerExecutionMessages": [],
"writerName": "html",
"writerVersion": "1.1.1",
"writerFormat": "html",
"writerPass": true,
"writerMessages" : [],
"writerOutput": "<!DOCTYPE html>\n<html lang=\"en\">\n..."
```



## PHP Example



```
1.
        <?php
2.
        function translate($json, $format='iso19115_2') {
3.
                //write to temp file to support cross-platform
                $temp = tmpfile();
5.
                fwrite($temp, $json);
6.
                fseek($temp, 0);
7.
                $meta = stream_get_meta_data($temp);
                exec("mdtranslator translate -o -w $format " . $meta['uri'], $out, $code);
8.
9.
                fclose($temp);
10.
11.
                $xml = empty($out) ? FALSE : json_decode($out[0]);
12.
13.
                if ($code > 0) {
14.
                    throw new \Exception("mdTranslator error.");
15.
                 } elseif (!is_object($xml) | !$xml->writerPass) {
                    throw new \Exception("JSON did not validate.");
16.
17.
18.
19.
                return $xml->writerOutput;
20.
21.
        ?>
```



# **Application Example**



ALCC Project Tracking System

mdJSON

mdTranslator

metadata. arcticlcc.org

ISO XML, mdJSON Webservice

### **Questions?**



github.com/adiwg/mdWorkshop
www.adiwg.org/mdTools
mdtranslator.adiwg.org
mdbook.adiwg.org



# ISO Developer's Toolkit - mdTranslator in Ruby

2015 CDI Workshop

May 11, 2015 Stan Smith, USGS



- 1. Complete the installation of Ruby and the adiwg-mdtranslator RubyGem if this is not already done.
- 2. Using your favorite code editor create a new file named 'translate.rb'.
- 3. Start by including the RubyGem 'adiwg-mdtranslator'.

```
# simple mdTranslator Ruby application
```

# include the adiwg-mdtranslator RubyGem in the application
require 'adiwg-mdtranslator'





4. Read in the file with your metadata content in mdJson format.

```
# read file in the mdJson file
my_file = File.open('C:\Users\...\full_example.json', 'r')
jsonObj = my_file.read
my_file.close
```

5. Call the mdTranslator.

```
# call mdtranslator with desired parameters
# mdtranslator uses a 'named' parameter list
metadata = ADIWG::Mdtranslator.translate(
    file: jsonObj, reader: 'mdJson', validate: 'normal',
    writer: 'iso19115_2', showAllTags: true)
```





6. Extract the output from the return.

```
# extract the metadata output from the returned metadata hash
if !metadata[:writerOutput].nil?
    writerOut = metadata[:writerOutput]
    metadata[:writerOutput] = 'Extracted'
end
```

7. Save the formatted metadata record.

```
# send the metadata output to a file
File.open('C:\Users\...\mdOutput.xml', 'w') { |file|
file.write(writerOut) }
```

8. All done!







#### 9. Other things to do ...

Examine other elements of the 'return hash'

```
# show metadata returned hash
require 'pp'
pp metadata
```

- Check return values for errors
- Handle error reporting
- Handle different writers
- Support mdTranslator.translate options
- Blah, blah, blah ...

### **Questions?**



github.com/adiwg/mdWorkshop
www.adiwg.org/mdTools
mdtranslator.adiwg.org
mdbook.adiwg.org



# ISO Developer's Toolkit - mdTranslator in Rails

2015 CDI Workshop

May 11, 2015 Stan Smith, USGS



- 1. From the command line, navigate to the directory to hold your new Rails application. This will become the parent directory of the new Rails application.
- 2. Generate the new rails application.
  - > rails new translator
- 3. A new directory named 'translator' will be created and filled with the new Rails application modules.
- 4. Test the new 'translator' rails application.
  - Change the directory to the Rails application just created
    - > cd translator
  - Start the Rails server.
    - > rails server
  - o From a browser type <a href="http://localhost:3000">http://localhost:3000</a>
  - You should see the 'Welcome aboard' page.



- 5. Generate a home page using a Rails helper.
  - > rails generate controller translate index
- 6. Test the new index page that Rails generated for you ... <a href="http://localhost:3000/translate/index">http://localhost:3000/translate/index</a>



## or maybe not!



- Found trouble on some Windows installations not sure of cause
- In file
  - .../app/views/layouts/application.html.erb comment out the line calling the javascript\_include\_tag so it looks like this:

```
<head>
 <title>mdTranslator API</title>
 <%= stylesheet_link_tag 'application', media: 'all', 'data-turbolinks-track' => true %>
 <!-- javascript_include_tag 'application', 'data-turbolinks-track' => true -->
 <%= csrf_meta_tags %>
</head>
```

This feature preserves head JavaScript from page to page as a means of saving load and compile time. Not need in this application anyway.



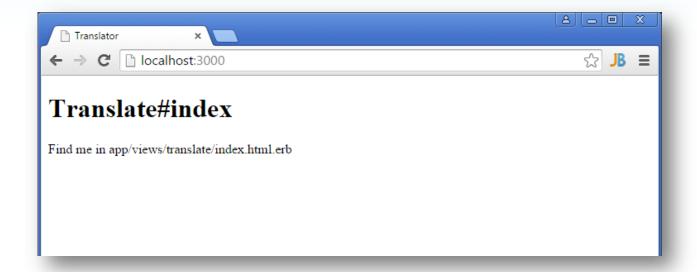


7. Next set the 'index' page as the default page for the website. In the .../config/routes.rb file set the root to 'translate#index'.

```
Rails. application. routes. draw do
   get 'translate/index'
   root 'translate#index'
```



8. Now the index page will display at root rather than 'Welcome aboard'.





9. Code a simple HTML form in the file .../app/views/translate/index.html.erb replacing all the Rails generated code.

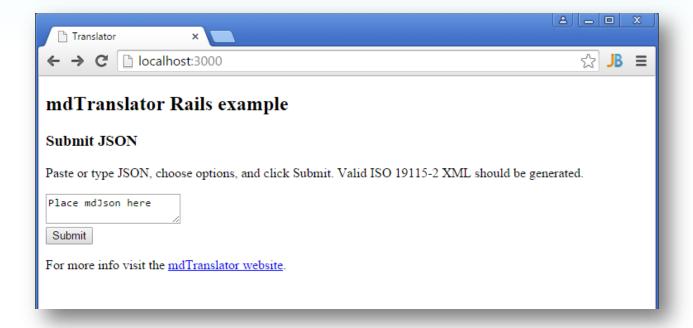


```
<!--
# Alaska Data Integration working group - ADIwg
# REST endpoint controller for demonstration of mdTranslator
<h2> mdTranslator Rails example</h2>
<div id="form container">
   <%= form_tag('/translate') do %>
      < di v >
         < h3 > Submit JSON < /h3 >
            Paste or type JSON, choose options, and click Submit.
            Valid ISO 19115-2 XML should be generated. <br/> <br/>
         <di v>
            <textarea name="file">Place mdJson here</textarea>
         </di v>
      </div>
      <div class="buttons">
         <i nput type="hi dden" name="form_i d" value="759352"/>
         <input id="saveForm" class="button_text" type="submit"</pre>
            name="submit" value="Submit"/>
      </div>
      <di v>
         >
            For more info visit the
            <a href="http://www.adiwg.org/mdTranslator">mdTranslator</a>
               websi te</a>.
         </di v>
   <% end %>
</di v>
```





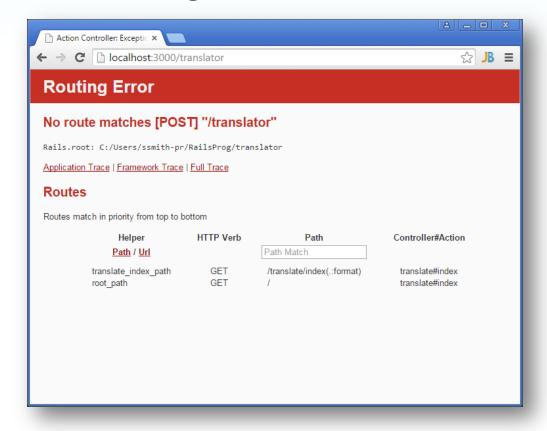
10. The updated index page now looks like this ...







11. Click 'Submit' and Rails shows a 'Routing Error'. We have not handled the HTTP POST in our routing or controller.





12. Rails only generated code in the router and controller to process the HTTP GET verb. Add a 'resources' statement to .../config/routes.rb to have Rails automatically handle all HTTP verbs, including POST.

```
Rails. application. routes. draw do
    get 'translate/index'
    root 'translate#index'
    resources: translate
```





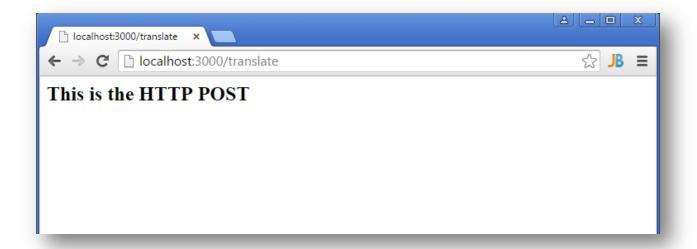
#### 13. Add code in the

.../app/controllers/translate\_controller.rb file intercept and process the HTTP POST request.

```
class TranslateController < ApplicationController
    # process GETs
    def index
   end
    # process POSTs
   def create
        render inline: '<h2>This is the HTTP POST</h2>'
    end
end
```



14. Now the HTTP POST is properly intercepted. When the 'Submit' button is pressed we see...







15. With the basic structure of the website complete connect the mdTranslator gem to the website. Add the adiwg-mdtranslator gem request to the .../Gemfile.

# Alaska Data Integration working group metadata translator gem ' adiwg-mdtranslator ' , '  $\sim>$  1.0 '

- 16. Update the website's gems. From the command line type:
  - > bundle update.
  - The adiwg-mdtranslator gem and all its dependencies will be loaded to your Rails website.
- 17. Remember to restart the Rails server after adding new gems.





18. Now we need to write a simple script to process the HTTP POST. In the ../app/controllers/translate\_controller.rb file replace the 'render inline:' statement we entered to test the connection and routing with something like this...

```
# process POSTs
def create

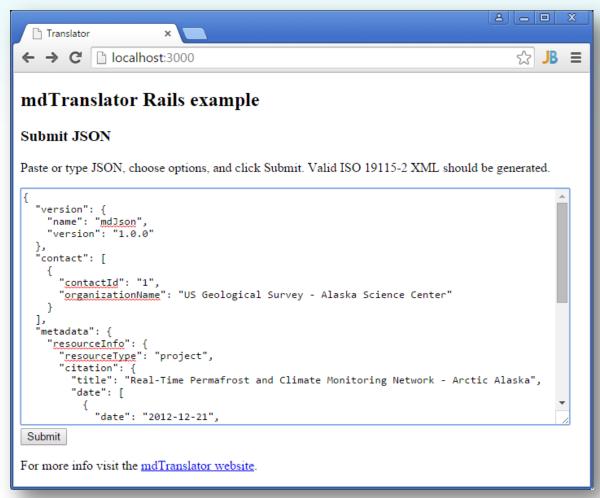
# load file and parameter from POST
fileObj = params[:file]

# call the ADIwg metadata translator
@mdReturn = ADIWG::Mdtranslator.translate(
    file: fileObj, reader: 'mdJson', validate: 'normal',
    writer: 'iso19115_2', showAllTags: false)

render xml: @mdReturn[:writerOutput]
end
```



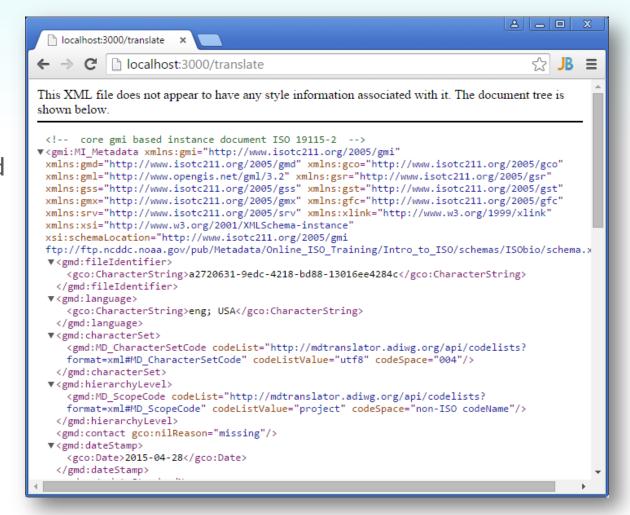
18. After the server restarts, navigate to the website root and enter some valid mdJson.







19. Click submit and you should see ISO 19115\_2 returned from you locally hosted web service.





• This 'non-award winning' website only demonstrates the simplicity of building a website that can interface with the ADIWG ISO Metadata Toolkit. A real website would need to be more robust checking for errors, handling all mdTranslator options, and handling all response types returned from the mdTranslator (XML. JSON, JSONp, text, plain). But it's a start.

### **Questions?**



github.com/adiwg/mdWorkshop
www.adiwg.org/mdTools
mdtranslator.adiwg.org
mdbook.adiwg.org